



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA

**RECURSO DIGITAL UTILIZANDO PIZARRÓN
INTERACTIVO PARA EL CÁLCULO DE LA DERIVADA
DE UNA FUNCIÓN**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO QUÍMICO

PRESENTA

FRANCISCO DAVID BARONA TOLEDO

DIRECTOR DE TESIS

I.Q. JOSE ANTONIO ZAMORA PLATA



Ciudad de México.

2016



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



i. JURADO.

ii. AGRADECIMIENTOS.

iii. DEDICATORIA.

iv. OBJETIVOS.

v. INDICE.



i. JURADO.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

FACULTAD DE ESTUDIOS
SUPERIORES "ZARAGOZA"

DIRECCIÓN

JEFE DE LA UNIDAD DE ADMINISTRACIÓN
ESCOLAR
PRESENTE.

Comunico a usted que al alumno(a) **Barona Toledo Francisco**, con número de cuenta **30804352-5** de la carrera **Ingeniería Química**, se le ha fijado el día **02** del mes de **Septiembre** de **2016** a las **11:00 horas** para presentar su examen profesional, que tendrá lugar en la sala de exámenes profesionales del Campus II de esta Facultad, con el siguiente jurado:

- | | |
|------------|---|
| PRESIDENTE | I.Q. TOMÁS VARGAS RAMÍREZ |
| VOCAL | I.Q. JOSÉ ANTONIO ZAMORA PLATA |
| SECRETARIO | M. EN I. CRESENCIANO ECHAVARRIETA ALBÍTER |
| SUPLENTE | I.Q. MARÍA ALEJANDRA VALENTÁN GONZÁLEZ |
| SUPLENTE | I.Q. BLAS MALDONADO SÁNCHEZ |

[Firma]
[Firma]
[Firma]
[Firma]
[Firma]

El título de la tesis que se presenta es: "Recurso digital utilizando pizarrón interactivo para el cálculo de la derivada de una función".

Opción de Titulación: Convencional

ATENTAMENTE
"POR MI RAZA HABLARÁ EL ESPÍRITU"
México, D. F. a 11 de Agosto de 2016.

DR. VÍCTOR MANUEL MENDOZA NUÑEZ
DIRECTOR

RECIBÍ:
OFICINA DE EXÁMENES PROFESIONALES
Y DE GRADO



ZARAGOZA
DIRECCIÓN
[Firma]
I.Q. DOMINGA ORTIZ BAUTISTA
JEFA DE LA CARRERA DE I.Q.



ii. AGRADECIMIENTOS.

A mis padres quienes han sido parte principal de mi formación profesional, ya que sin sus enseñanzas y su paciencia no hubiera podido llegar a lo que soy ahora.

A mis hermanos Daniela y Mauricio Barona por el apoyo incondicional y por su aliento diario para superar las cosas.

A Laura Elizabeth Laborde por el apoyo durante toda mi carrera por la paciencia, comprensión y por las enseñanzas dadas.

A mis amigos Andres, Gilberto, Denisse, Blanca, Carolina, Ivan, con quienes a pesar de ser mis amigos tuvieron la paciencia de ser mis maestros por lo cual siempre les estaré agradecido.

A todos mis profesores que me tuvieron como un alumno mas y que gracias a sus enseñanzas he podido lograr una gran meta que es ser un Ingeniero Químico.



iii. DEDICATORIA.

A mi familia quienes por ellos soy una persona profesionalista la cual pudo superar sus mayores retos. Para mis padres por su apoyo, comprensión, ayuda en los momentos difíciles y por ayudarme con los recursos necesarios durante mi etapa de estudiantil. A la UNAM por darme la oportunidad de ser parte de ella y por las experiencias aprendidas durante mi formación Universitaria.



vi. OBJETIVOS.

General

- ❖ Elaborar recurso digital utilizando pizarrón interactivo para el cálculo de la derivada de una función.

Particulares

- ❖ Describir los fundamentos del cálculo diferencial.
- ❖ Establecer una lista con las derivadas más comunes.
- ❖ Proponer algunos ejemplos de aplicación.
- ❖ Desarrollar un programa de computadora didactico que permita la memorización de las derivadas establecidas en el formulario.



v. INDICE

| | |
|--|----|
| RESUMEN..... | 1 |
| INTRODUCCIÓN..... | 2 |
| PLANTEAMIENTO DEL PROBLEMA..... | 3 |
| JUSTIFICACIÓN..... | 4 |
| MARCO TEORICO..... | 5 |
| CAPÍTULO 1 HERRAMIENTAS DIGITALES..... | 5 |
| 1.1 Tendencias que van a revolucionar el sistema educativo..... | 6 |
| 1.2 Técnicas visuales de aprendizaje..... | 7 |
| 1.2.1 ¿Qué es el aprendizaje visual?..... | 7 |
| 1.2.2 ¿Qué es un recurso digital?..... | 9 |
| 1.3 Pizarrones interactivos..... | 10 |
| 1.4 Elaboración de una lista de funciones y su derivada simple..... | 10 |
| CAPÍTULO 2 FUNDAMENTOS DEL CÁLCULO DIFERENCIAL..... | 17 |
| 2.1 Las matemáticas..... | 18 |
| 2.2 Historia del cálculo diferencial..... | 18 |
| 2.3 Cálculo diferencial..... | 19 |
| 2.4 Números reales..... | 20 |
| 2.5 Variable..... | 21 |
| 2.6 Función..... | 21 |
| 2.7 Derivada..... | 23 |
| CAPÍTULO 3 PROPUESTA DEL MATERIAL DIDÁCTICO..... | 24 |
| 3.1 Selección de funciones y sus derivadas..... | 25 |
| 3.2 Proceso estructural del material Didáctico..... | 25 |
| 3.2.1 Elaboración de ventana de ejemplos para el programa Learn Math..... | 26 |
| 3.2.2 Elaboración de ventanas de ejercicios para el programa Learn Math..... | 29 |
| 3.2.3 Elaboración de ventanas de conceptos para el programa Learn Math..... | 34 |
| 3.2.4 Elaboración de ventanas de presentación y menú de inicio del Learn Math..... | 37 |
| 3.3 Código de programación del material didáctico (Learn Math)..... | 39 |



| | |
|---|-----------|
| 3.3.1 Códigos de programación para los ejemplos de Learn Math..... | 39 |
| 3.3.2 Códigos de programación para los ejercicios de Learn Math. | 64 |
| 3.3.3 Códigos de programación para las ventanas de conceptos de Learn Math. | 81 |
| 3.3.4 Códigos de programación para las ventanas de inicio para Learn Math. | 88 |
| CAPÍTULO 4 PRUEBAS DEL MATERIAL DIDÁCTICO. | 95 |
| 4.1 Pruebas de ejecución del programa..... | 96 |
| 4.2 Pruebas del programa en distintas versiones de Windows ejecución del programa. | 96 |
| ANÁLISIS DE RESULTADOS..... | 97 |
| CONCLUSIONES..... | 98 |
| REFERENCIAS BIBLIOGRÁFICAS..... | 99 |



RESUMEN

En la presente tesis se elaboró una propuesta sobre un material didáctico utilizando un pizarrón interactivo. Esta propuesta se pretende utilizar como una herramienta de apoyo de los profesores para trabajar con los alumnos en clase de una manera visual e interactiva.

En la tesis se demuestra el procedimiento que se llevo a cabo para realizar el material interactivo. El material está realizado en Visual Studio 2012 y Matlab 2013, los cuales son herramientas útiles para la programación y creación de programas. Se utilizaron estos programas ya que son los programas utilizados durante la formación del Ingeniero Químico en la Facultad de Estudios Superiores Zaragoza.

El material didáctico contiene diferentes apartados que contienen teoría, ejercicios, y ejemplos propuestos por el autor de esta tesis, por lo cual realizo un formulario con diferentes derivadas, las imágenes fueron editadas y estructuradas en Photoshop CS6 debido a que las imágenes normales pueden verse pixeleadas o borrosas por eso se procedio a editar, ademas que el diseño del logotipo fue diseñado en el mismo Photoshop.

Esta propuesta trata de demostrar que con este tipo de herramientas interactivas, se pueden hacer más entendibles debido a la enseñanza visual, ademas de exortar a los alumnos a atreverse a hacer cosas mas competitivas y adecuadas a su época donde la tecnología es factor para el crecimiento del estudiante ya que entre mas tecnología sepa manejar mas productivo y competitivo puede ser.



INTRODUCCIÓN

La presente tesis está destinada a la elaboración de un recurso digital para las nuevas generaciones de la carrera de Ingeniería Química de la Facultad de Estudios Superiores Zaragoza. De acuerdo a los informes del Director, en la Facultad se tiene entendido que el índice de reprobación en la asignatura de matemáticas es demasiado alto por lo cual es necesario hacer algo por la comunidad estudiantil.

Analizando el plan de estudios y platicando con algunos profesores, se ha coincidido que los alumnos de nuevo ingreso llegan a la carrera mal preparados en matemáticas, no saben álgebra y por ende cálculo diferencial e integral; lo cual es preocupante porque la Ingeniería Química implica muchas matemáticas.

Además de que ingresan a la Facultad con esas deficiencias, a los alumnos y profesores se les hace muy extenso el programa de estudios de matemáticas I como de matemáticas II como para cubrirlo en tan poco tiempo. De esta manera, los profesores tienen que dar un tema por clase o incluso dos; esto puede ocasionar que los alumnos se saturen de información y más que razonar para asimilar los conceptos, terminan solamente mecanizando las operaciones matemáticas.

Se piensa que la tecnología hace que el ser humano se convierta en un simple usuario (no piensa, no razona y sólo aprieta botones). De acuerdo a la época en la que nos encontramos, las nuevas generaciones, están más familiarizadas con la tecnología y por esto, el realizar un recurso digital didáctico puede ser una herramienta viable para que aprendan a su manera nuevos temas.

En distintos países de distintos continentes la inversión sobre la educación y a su vez han podido observar que las nuevas generaciones piden aprender conforme a su época, esto quiere decir que piden medios de enseñanza tecnológicos conforme a su época lo cual en esos continentes les ha funcionado por lo que tienen estudiantes con un nivel superior al de otras escuelas en las cuales los métodos de enseñanza siguen siendo los mismos que el siglo pasado.

- **Material didáctico:** es aquel que reúne medios y recursos que facilitan la enseñanza y el aprendizaje. Suelen utilizarse dentro del ambiente educativo para facilitar la adquisición de conceptos, habilidades y actitudes¹⁸.



PLANTEAMIENTO DEL PROBLEMA.

La Ingeniería Química es la rama de la Ingeniería que se dedica al estudio, síntesis, desarrollo, diseño, operación y optimización de todos aquellos procesos industriales que producen cambios físicos, químicos y/o bioquímicos en los materiales. Los Ingenieros Químicos están involucrados en todas las actividades que se relacionen con el procesamiento de materias primas de origen animal, vegetal o mineral que tengan como fin obtener productos de mayor valor y utilidad. Sin embargo, parte esencial de la carrera de Ingeniería Química son las matemáticas ya que en toda la carrera las matemáticas son la base para cualquier diseño de equipo.

En la Facultad de Estudios Superiores Zaragoza se ha observado históricamente que el índice de reprobación de la materia de matemáticas es demasiado alto; por lo cual, es preocupante saber que los alumnos de nuevo ingreso no llegan preparados para poder tener un buen desempeño en sus materias. La mayoría tiene deficiencias tanto en álgebra como en trigonometría, cálculo diferencial e integral, por esta razón más del 60% reprueban y debido a eso, los alumnos se empiezan a atrasar en sus semestres y esto puede ocasionar que deserten de la carrera o que ya no terminen sus estudios en el tiempo establecido.



JUSTIFICACIÓN

En varios países de distintos continentes tienden a emplear la tecnología como un medio de aprendizaje. Los tiempos han cambiado las generaciones Millennials y Z exigen métodos de enseñanza muy diferentes a los que se usaban antes. Muchas personas creen que el uso de la computadora y el celular están acabando con los estudiantes sin embargo no se ponen a pensar que con la tecnología los jóvenes pasan mucho tiempo en contacto con los dispositivos de comunicación en donde pueden jugar e intercambiar mucha información a través de las redes sociales, y esto, se puede adecuar a las necesidades del profesor para dar un panorama más amplio sobre las cosas.

Algunas escuelas han adoptado técnicas de aprendizaje visuales utilizando programas de computadora debido a que estudios psicológicos demuestran que los alumnos aprenden más si les motiva con imágenes y juegos; se insiste en que de esta forma se les quedan más las cosas en sus mentes utilizando este tipo de aprendizaje, además de que es algo más adecuado a la época de los nuevos estudiantes.

Debido a esto, el propósito de esta tesis es crear un material visual para enseñarles a los nuevos estudiantes de esta época a utilizar un pizarrón digital interactivo sobre matemáticas. El software realizado es una muestra o demostración para aprender de una manera diferente sobre algunas derivadas básicas de matemáticas de una manera gráfica, visual, didáctica y actualizada. Esto con la finalidad de que el estudiante de matemáticas aprenda de una manera amigable, los conceptos de derivada de una función a través del software que aquí se proporciona. El alumno podrá ocupar el software en su casa y puede interactuar con él de una manera libre y cómoda.



CAPÍTULO 1

HERRAMIENTAS

DIGITALES





1.1 Tendencias que van a revolucionar el sistema educativo.

Las nuevas generaciones reclaman métodos de enseñanza más acordes con los perfiles que reclama un mercado laboral que no tiene nada que ver con lo vivido hasta ahora.¹³

La Asociación de Colegios Privados e Independientes (CICAE), en colaboración con Madrid Foro Empresarial y la Universidad Camilo José Cela (UCJC), han presentado el informe “**Nuevas tendencias en Educación**”, elaborado por el Observatorio de Tendencias de Coolhunting Community. En este trabajo se propone un decálogo de directrices emergentes e innovadoras para alcanzar la excelencia en el ámbito educativo¹³.

Power pupils (Empoderar a los alumnos)

Los alumnos son cada vez más conscientes de sus capacidades y talentos desde edades tempranas, y desarrollan habilidades como la confianza y la imaginación, el espíritu de emprendimiento, y la actitud activa hacia la formación. La inminente llegada de la Generación Z -que llega tras la llamada Generación X y los Millennials- plantea una nueva concepción del trabajo en la que la faceta profesional se desempeña a partir de una pasión, y no solo por la búsqueda de un empleo estable. La explicación de esta tendencia puede dividirse en tres apartados: «before is better» (fomentar el talento desde edades tempranas), “hackschooling” (acompañar a los pequeños para que descubran su vocación) y la «customización» (cursos modulares para que los alumnos puedan personalizar su educación en función de sus inquietudes educativas).¹³

B-Tech

También relacionada con la tecnología, su papel dentro de las organizaciones no solo ha transformado procesos, métodos y sistemas de trabajo, sino que también lo ha hecho con los modelos de negocio. Las aulas aplican estas herramientas para hacerlas más globales e hiperconectadas, crear documentos en la nube, buscar soluciones a problemas reales y obtener respuestas en tiempo real. El estudio concluye que, más allá de hacer uso de la tecnología como medio para conseguir llegar a un fin pedagógico, es necesario que el profesorado haga una reflexión para obtener su máximo aprovechamiento. Dentro de este punto, resultarán fundamentales recursos que permitan al alumno un feedback inmediato,



el uso del «big data» para personalizar contenidos al ritmo de aprendizaje de cada uno, y la posibilidad de integrarlos en distintos dispositivos.¹³

En un detallado estudio “Animation as an Aid to Multimedia Learning” (La animación como ayuda en el aprendizaje multimedia) que publicaba el “Educational Review Psychology” se mostraba la efectividad de la animación en estudiantes universitarios, a la hora de memorizar, atender, almacenar y recuperar información adquirida.¹³

1.2 Técnicas visuales de aprendizaje

Aprender a pensar. Aprender a aprender. Éstas son las habilidades esenciales para el éxito de los estudiantes. La investigación tanto en la psicología cognoscitiva como la teoría educativa nos indica que el aprendizaje visual es, entre muchos, el mejor método para enseñar a los estudiantes de todas las edades a pensar y la manera de aprender a aprender.¹⁴

1.2.1 ¿Qué es el aprendizaje visual?

Las técnicas de aprendizaje visual, maneras gráficas de trabajar con ideas y presentar información, enseñan a los estudiantes a ampliar sus pensamientos y a procesar, organizar y dar prioridad a la nueva información. Los diagramas visuales revelan patrones, interrelaciones e interdependencias. También estimulan el pensamiento creativo.¹⁴

Para los estudiantes que se expresan de manera visual y para aquellos que aprenden con más efectividad a través del uso de técnicas visuales, auditivas y otros multimedios, el aprendizaje visual inspira niveles más altos de logros. Estas técnicas de aprendizaje no necesitan muchos recursos, sólo papel y lápiz. Pero si tenemos a nuestra disposición recursos computacionales existen aplicaciones como SmartDraw® y Kidspiration® que facilitan y hacen más divertida la implementación de estas técnicas.¹⁴

Las técnicas de aprendizaje visual ayudan a los estudiantes a:



- **Aclarar pensamientos.** Los estudiantes ven cómo las ideas están conectadas y se dan cuenta de qué manera se puede agrupar u organizar la información. Con el aprendizaje visual, los conceptos nuevos se comprenden fácil y completamente.¹⁴
- **Reforzar la comprensión.** Los estudiantes recrean, en sus propias palabras, lo que han aprendido. Esto les ayuda a absorber e internalizar la nueva información, otorgándoles propiedad de sus ideas.¹⁴
- **Integrar nuevos conocimientos.** Al revisar los diagramas creados anteriormente y relacionarlos con los nuevos conocimientos, los estudiantes ven cómo los hechos y las ideas se acoplan.¹⁴
- **Identificar conceptos equivocados.** Como un mapa conceptual o de redes muestra lo que saben los estudiantes, los enlaces mal dirigidos o las conexiones incorrectas muestran lo que no comprenden.

Aprendizaje visual en el dominio de habilidades de pensamiento fundamentales: ¹⁴

- Ayudan a organizar las ideas
- Aumentan la creatividad
- La determinación de causa y efecto
- Haciendo la comparación
- Decodificación de las ideas
- Generación de preguntas
- La evaluación de la información
- Prueba de sus conocimientos

Aprendizaje visual allana el camino de la competitividad: ¹⁴

- Para resolver problemas
- Tomar decisiones
- Perfeccionar la memoria
- Mejorar la concentración
- Agilizar el pensamiento



Algunas técnicas de aprendizaje visual: ¹⁴

- Mapas conceptuales
- Línea del tiempo
- Mapas mentales o Diagramación mental
- Diagramas de causa y efecto
- Organizadores gráficos
- Diagramas de flujo
- Entre otros

1.2.2 ¿Qué es un recurso digital?

Un recurso digital es cualquier tipo de información que se encuentra almacenada en formato digital.¹⁷

De acuerdo con una definición formal del ISBD (ER) (International Estándar Bibliographic Description for Electronic Resources) (1997), entendemos por recurso digital todo material codificado para ser manipulado por una computadora y consultado de manera directa o por acceso electrónico remoto.

Los recursos digitales facilitan el almacenamiento, la organización y la recuperación de enormes cantidades de datos.¹⁷

A la estructura de información externa de un recurso digital se le denomina metadato y describe los aspectos técnicos y educativos del objeto. Siguiendo el estándar IEEE LOM (Learning Object Metadat) y ajustando el perfil de aplicación para el proyecto, se definen los siguientes metadatos obligatorios para los objetos de aprendizaje y se consideran opcionales el resto de campos de la especificación.¹⁷



1.3 Pizarrones interactivos.

Un pizarrón interactivo o pizarrón electrónico es un pizarrón blanco conectado a una computadora y a un proyector. ¹⁶



Es la manera más fácil de aprender ya que:

- ✓ Se podrán hacer clases más atractivas y dinámicas.
- ✓ Integrar imágenes y sonidos para que los alumnos participen más.
- ✓ El profesor, presentador o capacitador podrá repetir sesiones sin necesidad de escribir todo en el pizarrón una y otra vez ya que se puede guardar toda la información registrada en el pizarrón interactivo y reutilizarla.
- ✓ Es muy fácil de utilizar y de instalar.
- ✓ En un mundo donde los niños y jóvenes están a la vanguardia de la tecnología es importante brindarles una educación adecuada, ¡no hay porqué temerle a la tecnología!

Existen pizarrones interactivos o electrónicos que se pueden manejar con los dedos o cualquier objeto (sistema touch) y otros con plumillas electrónicas como si se tratara del ratón de una computadora. ¹⁶

1.4 Elaboración de una lista de funciones y su derivada simple.

En esta tesis se presenta una agrupación de diferentes tipos de funciones, se sabe que puede haber muchas maneras de agrupar las funciones dependiendo el tipo de función y cualquiera es correcta por lo que en esta tesis la agrupación seleccionada es la siguiente:

- ❖ Funciones constantes
- ❖ Funciones lineales



- ❖ Funciones polinomiales o polinómicas.
- ❖ Funciones logarítmicas
- ❖ Funciones trigonométricas

Se sabe que es una agrupación bastante básica sin embargo como se considera una propuesta inicial, se decidió trabajar con algo simple para verificar que con esta agrupación básica se puedan cumplir los objetivos planteados y si se cumple con los objetivos fomentar a los profesores o alumnos de la Facultad de Estudios Superiores Zaragoza a aumentar la dificultad del programa y hacerlo más complejo.



FORMULARIO PROPUESTO:

Funciones constantes y su derivada

$$y = c \quad ; \quad y' = 0$$

$$y = 5 \quad ; \quad y' = 0$$

$$y = 90 \quad ; \quad y' = 0$$

$$y = 76 \quad ; \quad y' = 0$$

$$y = 30 \quad ; \quad y' = 0$$

$$y = 16 \quad ; \quad y' = 0$$

$$y = 22 \quad ; \quad y' = 0$$

$$y = 0 \quad ; \quad y' = 0$$



Funciones lineales y su derivada

$$y = x \quad ; \quad y' = 1$$

$$y = 23x \quad ; \quad y' = 23$$

$$y = 67x \quad ; \quad y' = 67$$

$$y = 130x \quad ; \quad y' = 130$$

$$y = 297x \quad ; \quad y' = 297$$

$$y = 45x \quad ; \quad y' = 45$$

$$y = 3x \quad ; \quad y' = 3$$

$$y = 9x \quad ; \quad y' = 9$$

$$y = 7x \quad ; \quad y' = 7$$



Funciones polinómicas y su derivada

$$y = x^n \quad ; \quad y' = nx^{n-1}$$

$$y = x^4 \quad ; \quad y' = 4x^3$$

$$y = 3x^5 \quad ; \quad y' = 15x^4$$

$$y = 13x^{13} \quad ; \quad y' = 169x^{12}$$

$$y = x^{79} \quad ; \quad y' = 79x^{78}$$

$$y = x^0 \quad ; \quad y' = 0$$

$$y = x^2 \quad ; \quad y' = 2x$$

$$y = 5x^3 \quad ; \quad y' = 15x^2$$

$$y = 8x^4 \quad ; \quad y' = 32x^3$$



Funciones logarítmicas y su derivada

$$y = \ln x \quad ; \quad y' = \frac{1}{x}$$

$$y = 67 \ln 5x \quad ; \quad y' = \frac{67}{x}$$

$$y = \ln 78x \quad ; \quad y' = \frac{1}{x}$$

$$y = \ln 150x \quad ; \quad y' = \frac{1}{x}$$

$$y = 2 \ln 2x \quad ; \quad y' = \frac{2}{x}$$

$$y = 5 \ln 3x \quad ; \quad y' = \frac{5}{x}$$

$$y = 13 \ln 2x \quad ; \quad y' = \frac{13}{x}$$

$$y = 17 \ln 9x \quad ; \quad y' = \frac{17}{x}$$

$$y = 30 \ln 15x \quad ; \quad y' = \frac{30}{x}$$



Funciones trigonométricas y su derivada

$$y = \tan x \quad ; \quad y' = \sec^2 x$$

$$y = \cos x \quad ; \quad y' = -\sin x$$

$$y = \sin x \quad ; \quad y' = \cos x$$

$$y = \sin 15x \quad ; \quad y' = 15 \cos 15x$$

$$y = 2 \cos 11x \quad ; \quad y' = -22 \sin 11x$$

$$y = 34 \tan 69x \quad ; \quad y' = 34 \cdot 69 \sec^2 69x$$

$$y = 2 \sin 11x \quad ; \quad y' = 22 \cos 11x$$

$$y = 2 \sin 48x \quad ; \quad y' = 96 \cos 48x$$

$$y = 5 \cos 33x \quad ; \quad y' = -165 \sin 33x$$

$$y = 2 \tan 7x \quad ; \quad y' = 14 \sec^2 7x$$

$$y = 17 \tan 3x \quad ; \quad y' = 51 \sec^2 3x$$

$$y = \sin 20x \quad ; \quad y' = 20 \cos 20x$$



CAPÍTULO 2

FUNDAMENTOS DEL CÁLCULO DIFERENCIAL.





2.1 Las matemáticas

La matemática es la ciencia deductiva que se dedica al estudio de las propiedades de los entes abstractos y de sus relaciones. Esto quiere decir que las matemáticas trabajan con números, símbolos, figuras geométricas, etc.

Además de lo expuesto no podemos pasar por alto que existen dos importantes tipos de matemáticas:

- ✓ Las matemáticas puras, que se encargan de estudiar la cantidad cuando está considerada en abstracto.
- ✓ Las matemáticas aplicadas, que proceden a realizar el estudio de la cantidad pero siempre en relación con una serie de fenómenos físicos.⁷

2.2 Historia del cálculo diferencial

Newton y Leibniz

El cálculo Diferencial e Integral ha sido reconocido como el instrumento más efectivo para la investigación científica que jamás hayan producido las matemáticas. Concebido para el estudio del cambio, el movimiento y la medición de áreas y volúmenes, el cálculo es la invención que caracteriza la revolución científica del siglo XVII. Su creación se debe al trabajo independiente de dos matemáticos, el inglés Isaac Newton (1642-1727) y el alemán Gottfried Wilhelm Leibniz (1646-1716), quienes publicaron sus investigaciones entre los años de 1680 y 1690. Leibniz en 1684, en la revista Acta Eruditorum, y Newton en 1687, en su gran obra Principia Mathematica Philosophiae Naturalis.¹⁵



Sir Isaac Newton
(1642–1727)



Gottfried Wilhelm Leibniz
(1646–1716)

Fig.1.- Issac newton y Gotfried Leibniz Participante principales del descubrimiento del cálculo diferencial



Isaac Newton hijo de granjeros, nació en Lincolnshire, Inglaterra, en el día de Navidad de 1642 y llegó en 1669 a ocupar, en la Universidad de Cambridge, la Cátedra Lucasiana como profesor de matemáticas. En la presentación de sus ideas, Newton recurre a argumentos basados en el movimiento y la dinámica de los cuerpos. Así, las variables son vistas como algo que cambia o fluye con el tiempo (fluente) y a su derivada o razón de cambio con respecto al tiempo la llama su fluxión. El problema básico del cálculo es, para Newton, el estudio de las relaciones entre fluentes y sus fluxiones. En 1671, Newton concluye su tratado sobre el método de fluxiones que no es publicado sino hasta 1736, casi diez años después de su muerte, ocurrida en 1727.¹⁵

G. W. Leibniz hijo de un profesor de filosofía, nació en la ciudad de Leipzig, Alemania, en 1646. Ingresó a la universidad a la edad de quince años y obtuvo el doctorado en filosofía a la edad de 21 años. El interés de Leibniz por las matemáticas nació en 1672 durante una visita a París, donde el matemático holandés Christiaan Huygens (1629-1695) lo introdujo al estudio de la teoría de curvas.¹⁵

Leibniz se interesó en las cuestiones de lógica y de notación para la investigación formal, y su cálculo infinitesimal es el ejemplo supremo, en todas las ciencias y las matemáticas, de un sistema de notación y terminología perfectamente adaptado a su objeto de estudio. En el sentido anterior, Leibniz formalizó, con su notación, las propiedades y reglas fundamentales de los procesos de derivación e integración, haciendo de su aplicación a los más variados problemas, un ejercicio de rutina que un estudiante puede aprender desde sus primeros años.¹⁵

2.3 Cálculo diferencial.

El Cálculo Diferencial, es una parte importante del análisis matemático y dentro del mismo del cálculo infinitesimal. Consiste en el estudio del cambio de las variables dependientes cuando cambian las variables independientes de las funciones o campos objetos del análisis. El principal objeto de estudio en el cálculo diferencial es la derivada. Una noción estrechamente relacionada es la de Diferencial de una función.²

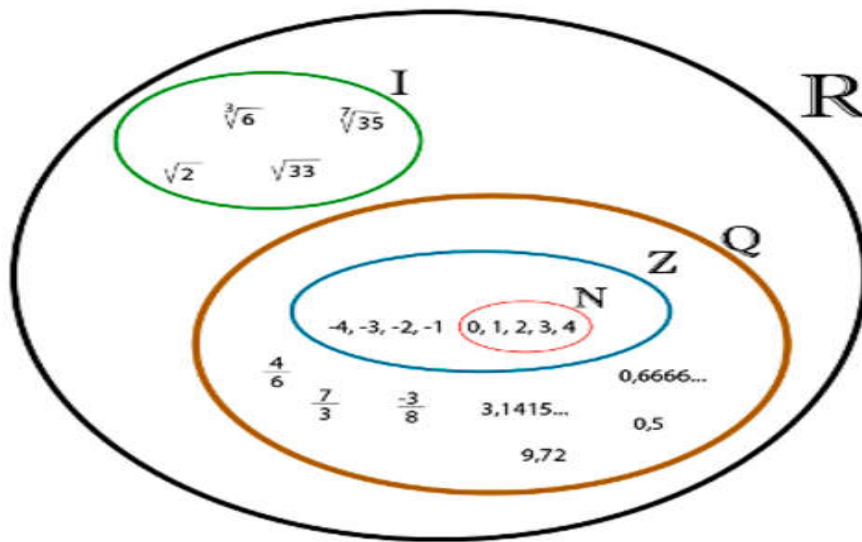
En el estudio del cambio de una función cuando cambian sus variables independientes es de especial interés para el cálculo diferencial, en el que el cambio de las variables es infinitesimal, esto es, cuando dicho cambio tiende a cero (se hace tan pequeño como se



deseo). Y es que el cálculo diferencial se apoya constantemente en el concepto básico del límite. El paso al límite es la principal herramienta que permite desarrollar la teoría del cálculo diferencial y la que lo diferencia claramente del álgebra.⁸

2.4 Números reales.

El conjunto de los números reales pertenece en matemáticas a la recta numérica que comprende a los números racionales y a los números irracionales. Esto quiere decir que incluyen a todos los números positivos y negativos, el símbolo cero, y a los números que no pueden ser expresados mediante fracciones de dos enteros que tengan como denominador a números no nulos (excluye al denominador cero).^{9,3}



- N= números naturales (enteros positivos)
- Z= números enteros (positivos y negativos)
- Q= números racionales (fracciones y decimales)
- I= irracionales

Fig.2.- Representación grafica de lo que son los números reales.³

El sistema de los números reales es la estructura algebraica adecuada al propósito del cálculo diferencial e integral. Son precisamente los atributos y las relaciones expresables en términos de este tipo de números, los objetos de estudio de esa rama de las matemáticas. Las propiedades especiales del sistema de los números reales permiten definir los conceptos fundamentales para la descripción y estudio del cambio y el movimiento.³

La propiedad que distingue al sistema de los números reales del sistema de los números racionales es la propiedad de continuidad. Esta propiedad, de carácter geométrico o



topológico, es la que permite dar un sentido preciso a los conceptos fundamentales de límite y continuidad, sobre los cuales se desarrolla el cálculo diferencial e integral.^{9,15}

2.5 Variable.

Una variable es una propiedad o un atributo que puede tomar uno o varios valores dados por los elementos de un conjunto. Este conjunto constituido por todos los elementos o variables, que pueden sustituirse unas a otras es el universo de variables. Se llaman así porque varían, y esa variación es observable y medible.

- **Variables cuantitativas**, cuando se expresan en números, como por ejemplo la longitud o el peso.
- **Variables cualitativas**, expresan cualidades, por ejemplo, designar con letras las preferencias de los estudiantes por sus materias de estudio.
- **Variables continuas**, son las que pueden tener cualquier valor como el peso o la altura.
- **Variables discontinuas**, son las que tienen valores determinados, como por ejemplo, la cantidad de hijos de una familia.
- **Variables dependientes**, que constituyen el objeto de investigación, por ejemplo crecimiento y desarrollo de los niños, son las que se ven modificadas por las independientes.¹⁰

2.6 Función.

En matemática se denominan funciones a la relación entre dos magnitudes, con correspondencia de valores. Por un lado hay un conjunto de variables que se llaman independientes, y por otro el de las variables dependientes, relacionadas de alguna manera con el primer conjunto. Por ejemplo si el primer conjunto estaría formado por los números 2, 4, 6 y 8 y el segundo, por los números 4, 16, 36 y 64, éstos serían los cuadrados de los primeros. Cada elemento del primer conjunto se relaciona con un único elemento del segundo. Las funciones pueden graficarse en coordenadas mediante puntos que representan a los dos tipos de variables.¹¹

Una función real de variable real f es una regla de correspondencia f entre dos variables reales x y y , tal que a cada valor de la primera corresponde, según esa regla, un único valor



de la segunda. Al dominio de la variable independiente x se le llama el dominio de la función y al dominio de la variable dependiente y se le llama el contra dominio de la función.¹⁵

A las funciones reales de variable real definidas en toda la recta y de la forma.¹⁵

$$f(x) = x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$$

Donde a_1, \dots, a_n son números reales, se les llama funciones polinomiales.¹⁵

En los casos particulares:

$$f(x) = ax + b \text{ con } a, b \in \mathbb{R},$$

la función se llama función lineal "y".¹⁵

$$f(x) = ax^2 + bx + c \text{ con } a, b, c \in \mathbb{R},$$

la función se llama función cuadrática.¹⁵

Así como las Funciones Logarítmicas y Trigonométricas.



2.7 Derivada.

La derivada en un punto de funciones de una variable (derivada total usual). Como es sabido, una derivada de una función $f(x)$ en un punto x_0 se obtiene comparando el valor de la función en ese punto y en otro muy próximo. Bajo ciertas condiciones, que $f(x)$ debe cumplir en entorno de x_0 , el cociente entre la diferencia de valores de la función y la diferencia de valores de la variable tiende a un límite, que llamamos derivada. Junto a la figura se indican varias notaciones usuales para la derivada calculada. El valor de la derivada en el punto coincide con la pendiente (tga) de la recta tangente a la función en ese punto.¹²

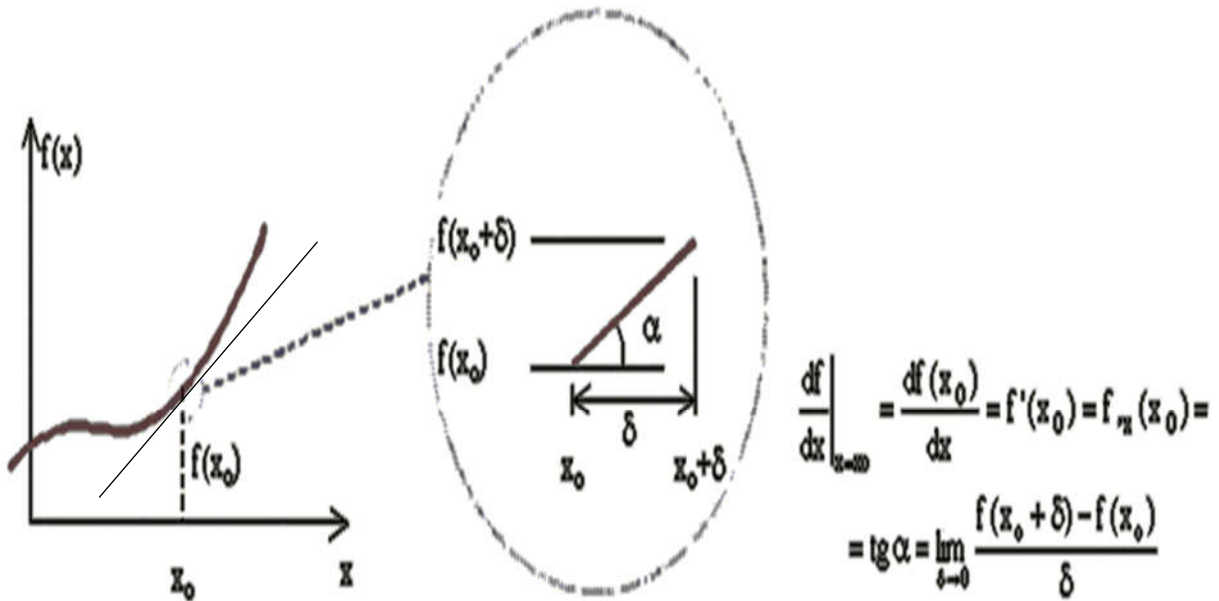
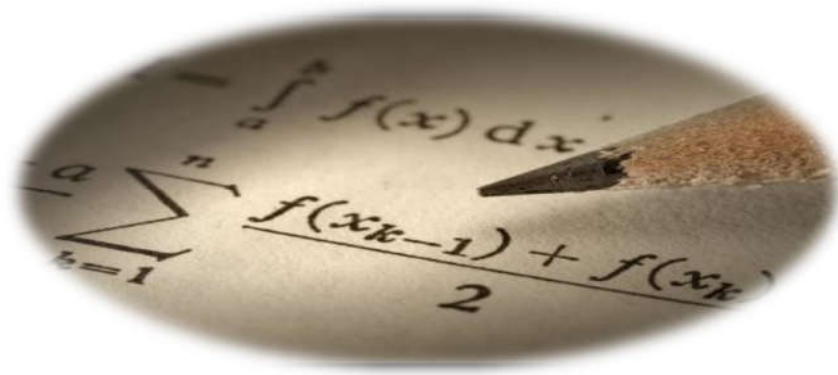


Fig.3.- Concepto de derivada de manera gráfica.¹²



CAPÍTULO 3

PROPUESTA DEL MATERIAL DIDÁCTICO.





3.1 Selección de funciones y sus derivadas.

El material didáctico contienen ejemplos de derivadas propuestas por el alumno, además los números ocupados en el programa Learn Math no son muy comunes en los ejercicios, esto con la finalidad de darle un poco más de variedad a comparación de que algunos libros a veces traen las mismas funciones y derivadas.

3.2 Proceso estructural del material Didáctico.

Lo primero que se realizó fue analizar como podría ser la mejor distribución en las ventanas del material para que este fuera amigable y sencillo. Por lo cual se optó por hacer una ventana exclusiva para ejercicios, una ventana exclusiva para ejemplos y a su vez una ventana más para un poco de teoría. En la ventana de teoría, se tendrán solamente conceptos simples que den una idea al alumno o la persona que lo utilice, ya que el contenido completo se encuentra en los libros.

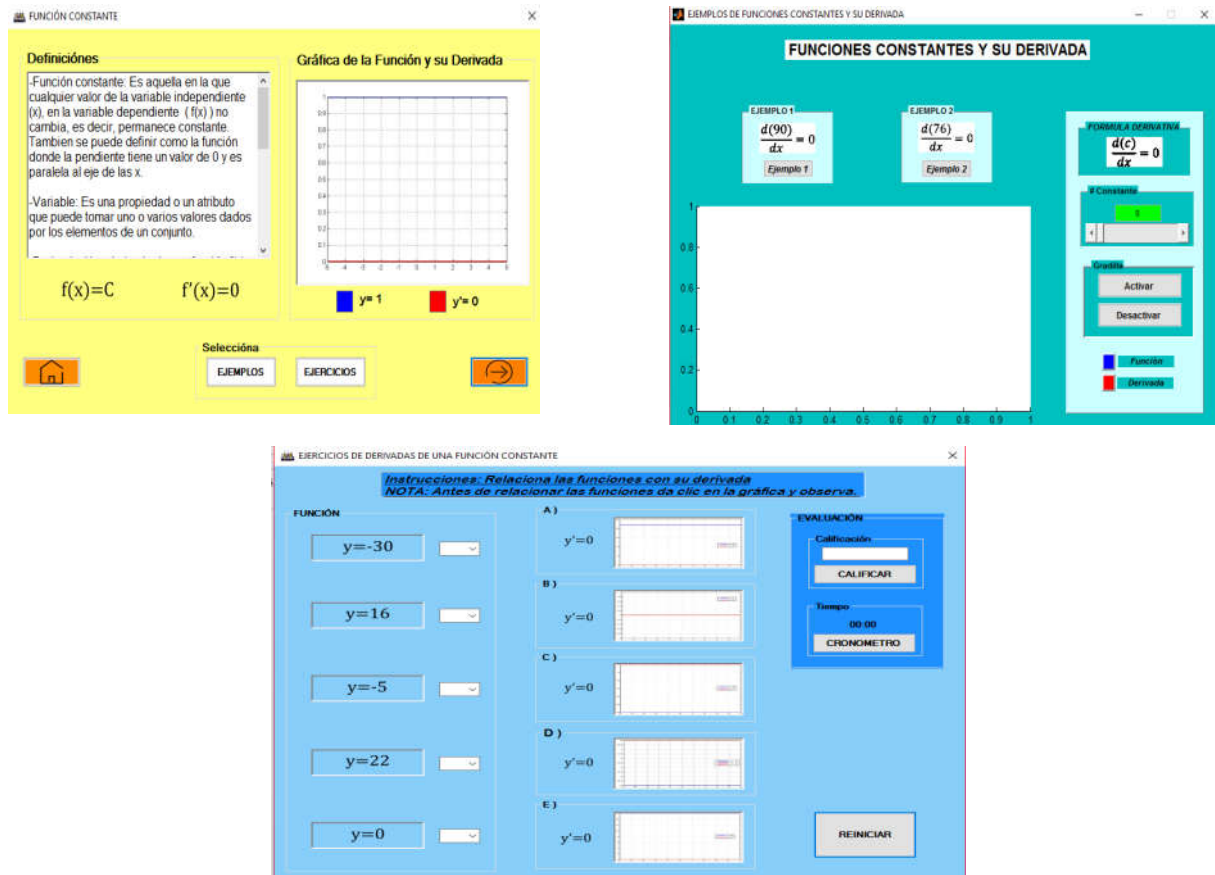


Fig.4.- Visualización de diferentes ventanas del programa Learn Math.



3.2.1 Elaboración de ventana de ejemplos para el programa Learn Math.

Una vez que se armó la estructura del diseño de las ventanas, lo que prosiguió fue realizar la ventana que contendría ejemplos para poder demostrar a las personas de una manera más gráfica como varía el comportamiento de las líneas conforme cambia el tipo de función y a su vez poder observar el comportamiento de la derivada de cada función.

Se propusieron 5 ventanas con ejemplos sencillos de derivación para poder observar de una manera más sencilla la derivada sin necesidad de escribir o graficar a mano.

Los ejemplos son completamente interactivos ya que con solo un clic se seleccionan el ejemplo de mayor agrado y de esta manera interactuar con los botones que aparecen en la ventana, además de verificar que ocurre con la gráfica.

Las funciones seleccionadas para estos ejemplos son:

- ✓ Ejemplos de funciones constantes y su derivada:

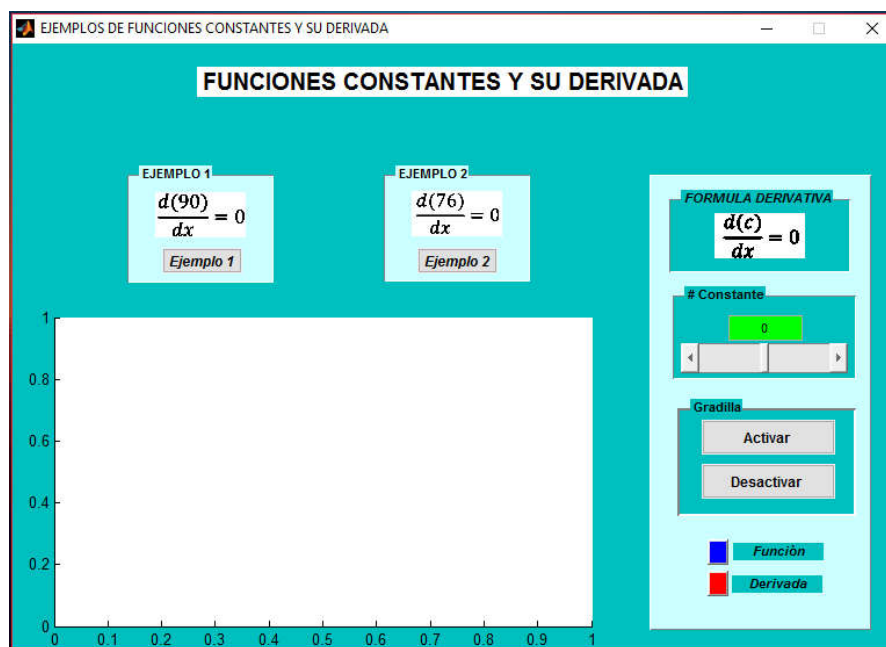


Fig 5.- Ventana de ejemplos de funciones constantes.



✓ Ejemplo de funciones lineales y su derivada.

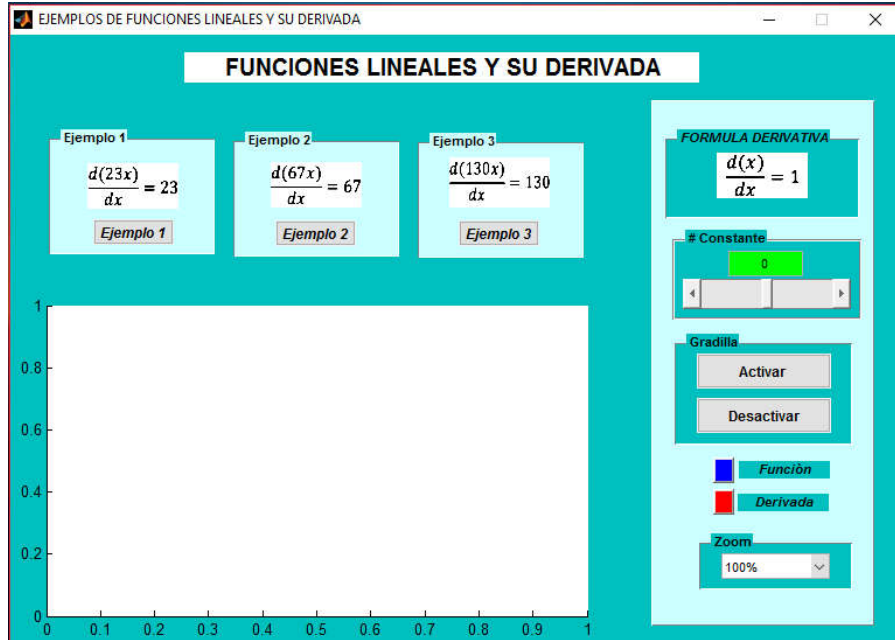


Fig. 6.- Ventana de ejemplos de funciones lineales.

✓ Ejemplos de funciones polinomiales y su derivada.

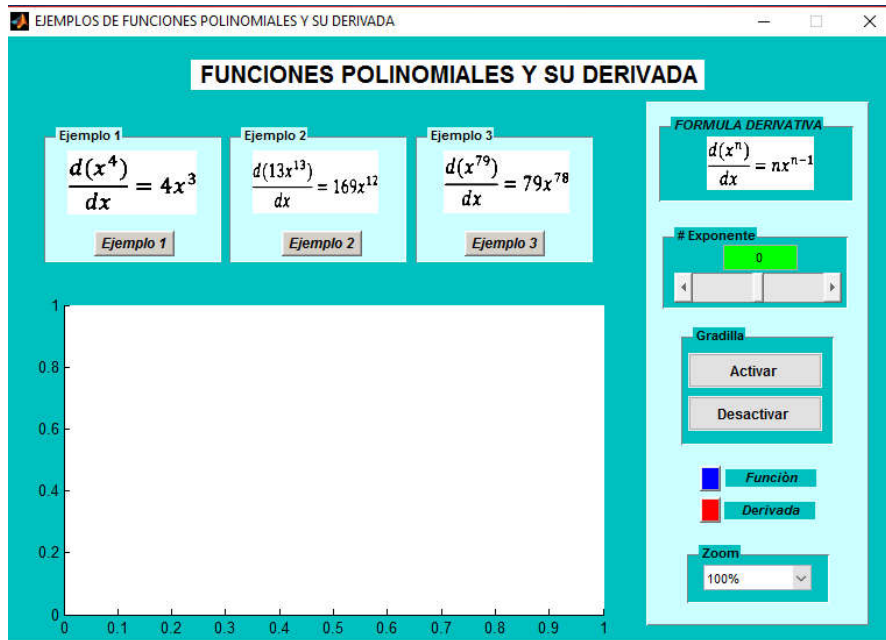


Fig. 7.- Ventana de ejemplos de funciones polinomiales.



- ✓ Ejemplos de Funciones logarítmicas y su derivada.

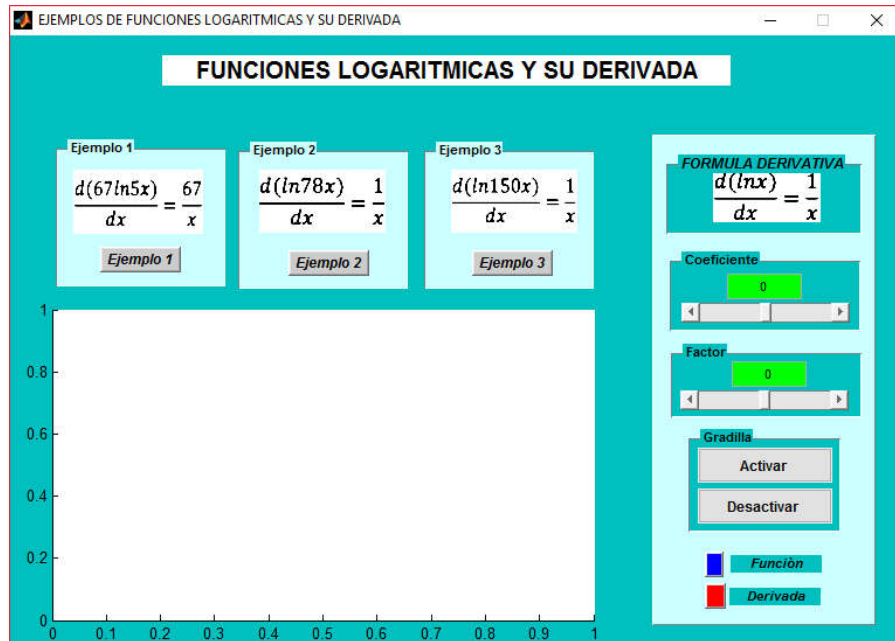


Fig. 8.- Ventana de ejemplos de funciones logarítmicas.

- ✓ Ejemplos de funciones trigonométricas y su derivada.

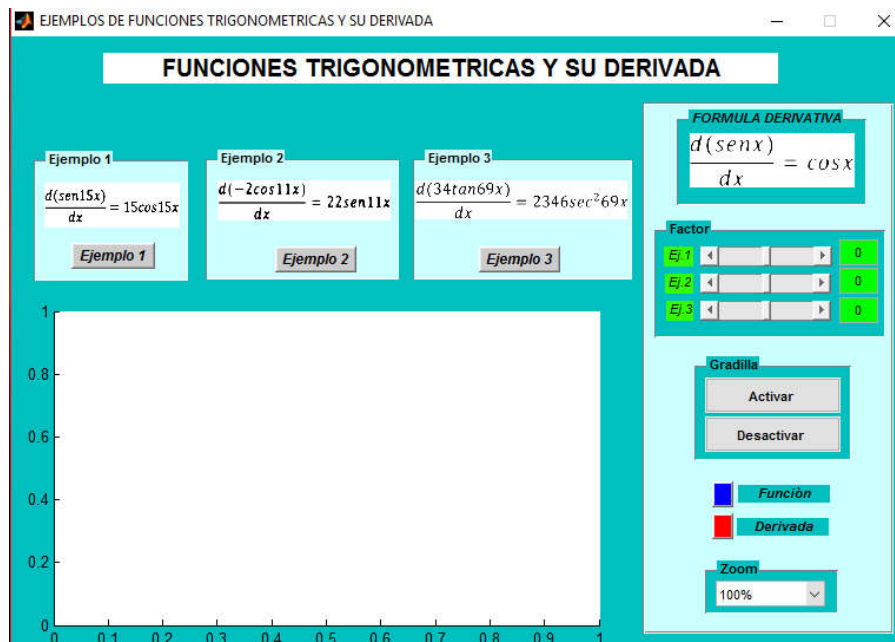


Fig. 9.- Ventana de ejemplos de funciones trigonométricas.



Los ejemplos fueron realizados por el alumno, el cual trató de buscar números poco comunes para llevarle al alumno o a las personas una manera de aprender diferente a comparación a la de los ejercicios que se encuentran en los libros.

Estos ejemplos fueron programados en la Herramienta Simulink de Matlab ya que con esta herramienta podemos realizar graficas con un código de programación que el Visual Studio no tiene, por ese motivo es que decidió ocupar esta herramienta del Matlab, además de ser una herramienta útil para hacer programas interactivos tal y como lo es requerido en esta tesis.

3.2.2 Elaboración de ventanas de ejercicios para el programa Learn Math.

Para el programa interactivo se determinó que debería de tener una opción donde se abriera una ventana en la cual aparecieran algunos ejercicios para seleccionar la respuesta correcta esto con la finalidad de hacer que el programa cumpla con el objetivo de ser interactivo y amigable con el estudiante.

Los ejercicios fueron seleccionados por el autor de esta tesis y se propusieron 5 ejercicios por cada tema propuesto.

Al igual que los ejemplos, los ejercicios para los temas seleccionados son:

- ✓ Ejercicios de derivadas de una función constante.

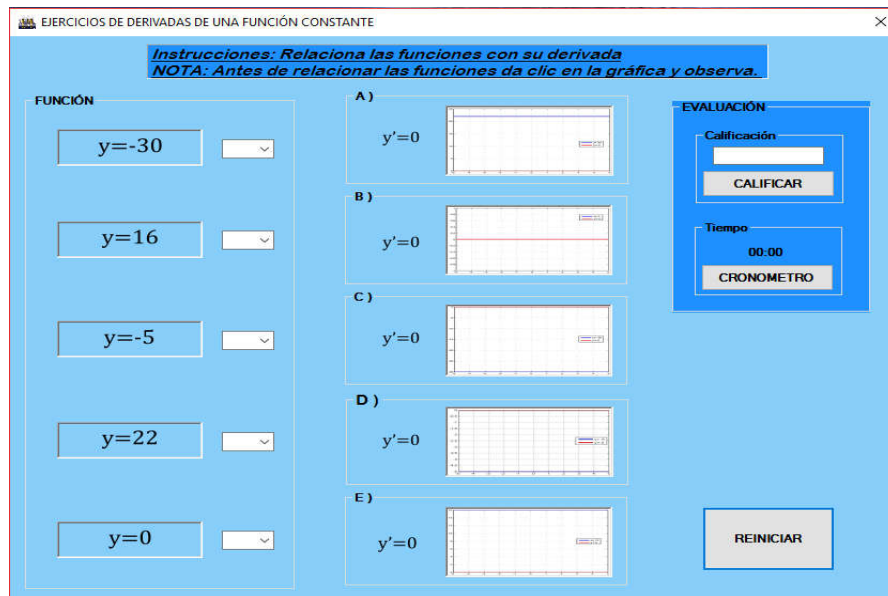


Fig. 10.- Ventana de ejercicios de funciones constantes.



✓ Ejercicios de derivadas de una función lineal.

EJERCICIOS DE DERIVADA DE UNA FUNCIÓN LINEAL

Instrucciones: Relaciona las funciones con su derivada.
NOTA: Al acertar en la Relación, da clic en la imagen para poder ver su gráfica.

FUNCIÓN

$y=45x$

$y=297$

$y=3x$

$y=9x$

$y=7x$

EVALUACIÓN

Calificación

CALIFICAR

Tiempo

00:00

CRONOMETRO

REINICIAR

A) $y'=297$

B) $y'=3$

C) $y'=45$

D) $y'=7$

E) $y'=9$

Fig. 11.- Ventana de ejercicios de funciones lineales.

✓ Ejercicios de derivadas de una función polinómica.

EJERCICIOS DE DERIVADA DE UNA FUNCIÓN POLINOMIAL

Instrucciones: Relaciona las funciones con su derivada.
NOTA: Al acertar en la Relación, da clic en la imagen para poder ver su gráfica.

FUNCIÓN

$y=x^0$

$y=x^2$

$y=-3x^5$

$y=5x^3$

$y=8x^4$

EVALUACIÓN

Calificación

CALIFICAR

Tiempo

00:00

CRONOMETRO

REINICIAR

A) $y'=15x^2$

B) $y'=-15x^4$

C) $y'=2x$

D) $y'=32x^3$

E) $y'=0$

Fig. 12.- Ventana de ejercicios de funciones polinomiales.



✓ Ejercicios de derivadas de una función logarítmica.

EJERCICIOS DE DERIVADAS DE UNA FUNCIÓN LOGARÍTMICA

*Instrucciones: Relaciona las funciones con su derivada.
NOTA: Al acertar en la Relación, da clic en la imagen para poder ver su gráfica.*

FUNCIÓN

- $y=2\ln(2x)$
- $y=5\ln(3x)$
- $y=13\ln(2x)$
- $y=17\ln(9x)$
- $y=30\ln(15x)$

A) $y'=2/x$

B) $y'=30/x$

C) $y'=17/x$

D) $y'=5/x$

E) $y'=13/x$

EVALUACIÓN

Calificación

CALIFICAR

Tiempo

00:00

CRONOMETRO

REINICIAR

Fig. 13.- Ventana de ejercicios de funciones logarítmicas.

✓ Ejercicios derivadas de una función trigonométrica.

EJERCICIOS DE DERIVADAS DE FUNCIONES TRIGONOMETRICAS

*Instrucciones: Relaciona las funciones con su derivada.
NOTA: Al acertar en la Relación, da clic en la imagen para poder ver su gráfica.*

FUNCIÓN

- $y=2\text{sen}(48x)$
- $y=5\text{cos}(33x)$
- $y=-2\text{tan}(7x)$
- $y=17\text{tan}(3x)$
- $y=\text{sen}(20x)$

A) $y'=-165\text{sen}(33x)$

B) $y'=-14\text{sec}^2(7x)$

C) $y'=51\text{sec}^2(3x)$

D) $y'=96\text{cos}(48x)$

E) $y'=20\text{cos}(20x)$

EVALUACIÓN

Calificación

CALIFICAR

Tiempo

00:00

CRONOMETRO

REINICIAR

Fig. 14.- Ventana de ejercicios de funciones trigonométricas.



Como se puede observar en las figuras (9,10,11,12,13). En el programa habrá 5 ventanas con ejercicios de diferente tipo de funciones, en cada ventana al dar clic en el botón calificar si la calificación es aprobatoria aparecerá un mensaje de que se ha aprobado el ejercicio tal y como se muestra en la figura 14, además de que las respuestas correctas se marcarán de un color amarillo.



Fig. 15.- Ventana de mensaje emergente de aprobación.

Sin embargo si la calificación es reprobatoria aparecerá un mensaje de que no se ha aprobado el ejercicio y las respuestas incorrectas serán marcadas con color rojo.



Fig. 16.- Ventana de mensaje emergente de reprobación.



Como se puede observar en las figuras anteriores el programa es didáctico, y entretenido a pesar de ser sencillo.

En la ventana hay cosas que también podrían ser útiles como por ejemplo:

Al dar clic en las gráficas de los ejemplos, la imagen se abrirá en otra ventana para facilitar su manipulación en caso de que se pudieran hacer algunas aclaraciones en el pizarrón. (Ver figura 16).

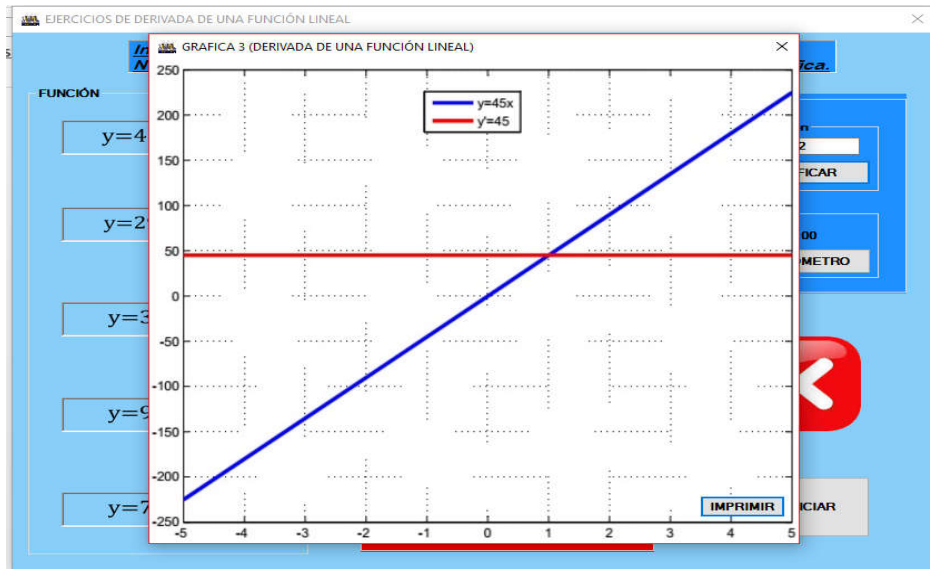


Fig. 17.- Ventana emergente de expansión de gráfica.

Las gráficas aumentadas emergentes que aparecen al dar clic en las gráficas del ejercicio pueden imprimirse, en dado caso que sea necesario, por ello tiene un botón de impresión que manda a imprimir la gráfica de manera automática.

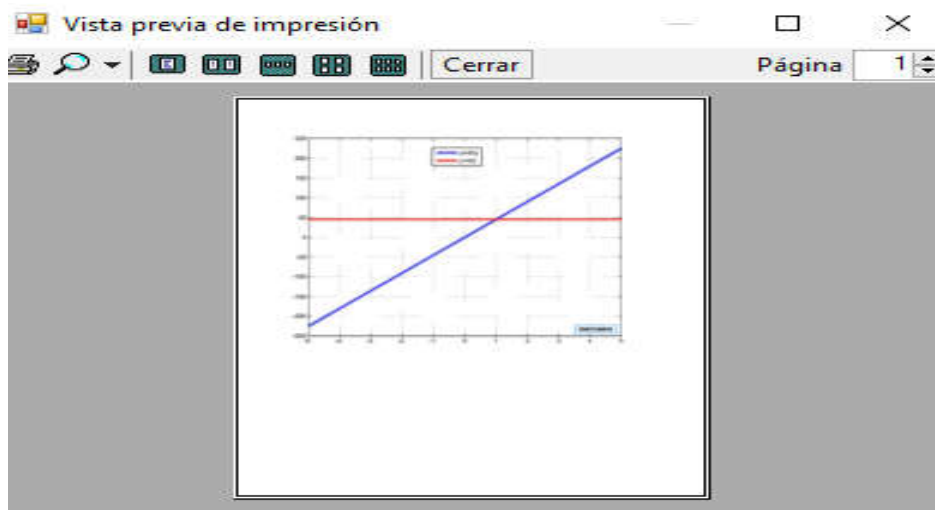


Fig. 18.- Ventana emergente para impresión de gráfica.



Además como una acción complementaria se tiene un botón con el cual se puede activar la función de cronometrar el tiempo, esto para medir el tiempo que cada uno se tardaría en contestar el ejercicio y así el profesor poder identificar cuales son los alumnos que les cuesta mas trabajo aprender y poder dedicar un poco mas de tiempo a aquellos que lo necesiten, ya cuando se haya terminado de responder el ejercicio dar clic en el botón calificar automáticamente se detendrá el cronómetro para checar el tiempo que se tardó en responder. (Ver figura 18)

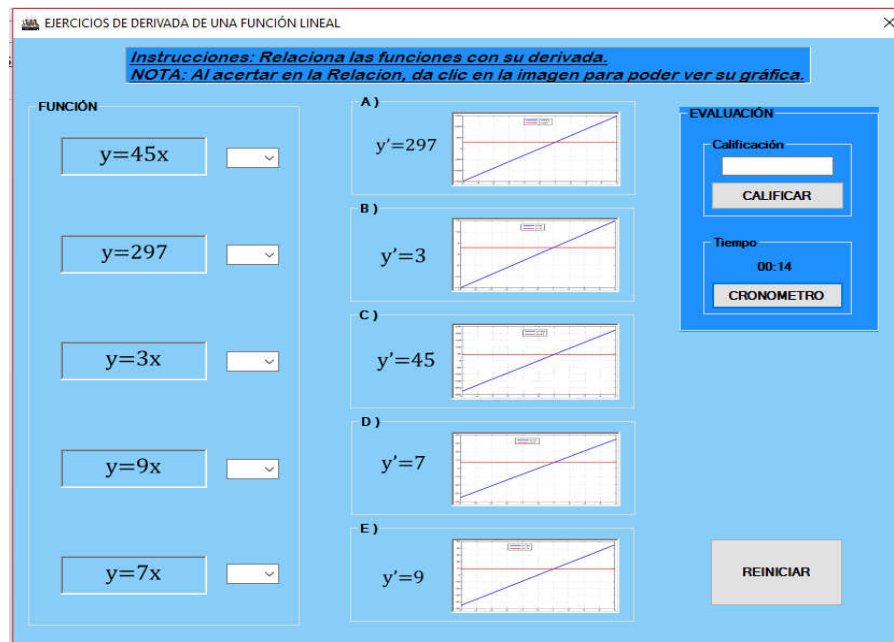


Fig. 19.- apartado para cronometro de uso opcional.

3.2.3 Elaboración de ventanas de conceptos para el programa Learn Math.

Para cada tipo de función en este programa se propuso una estructura de una ventana la cual contendría algunos conceptos básicos y sencillos para poder entender un poco sobre las funciones y su derivada, esta ventana contiene una sección donde se encuentran conceptos más aparte las fórmulas de derivación sencillas o inmediatas de una función además de un ejemplo gráfico sencillo para poder dar una idea al estudiante de como es el comportamiento de la gráfica de esa función. Además desde esa ventana tiene botones en los cuales al dar clic se puede enlazar a los ejercicios propuestos o a los ejemplos propuestos sin necesidad de regresar al menú de inicio.



- ✓ Ventana de conceptos básicos de una función constante y su derivada.

Definiciones

-Función constante: Es aquella en la que cualquier valor de la variable independiente (x), en la variable dependiente (f(x)) no cambia, es decir, permanece constante. También se puede definir como la función donde la pendiente tiene un valor de 0 y es paralela al eje de las x.

-Variable: Es una propiedad o un atributo que puede tomar uno o varios valores dados por los elementos de un conjunto.

$f(x)=C$ $f'(x)=0$

Gráfica de la Función y su Derivada

$y=1$ $y'=0$

Seleccióna
EJEMPLOS EJERCICIOS

Fig. 20.-Ventana para opción de conceptos de funciones constantes de Learn Math

- ✓ Ventana de conceptos básicos de una función lineal y su derivada.

Definiciones

-Función lineal: Es una función cuyo dominio son todos los números reales, cuyo codominio también todos los números reales, y cuya expresión analítica es un polinomio de primer grado. La función lineal se define por la ecuación $f(x)= mx + b$ ó $y = mx + b$ llamada ecuación canónica, en donde m es la pendiente de la recta y b es el intercepto con el eje Y.

-Variable: Es una propiedad o un atributo que puede tomar uno o varios valores dados

$f(x)=mx+b$ $f'(x)=m$

Gráfica de la Función y su Derivada

$y=3x+2$ $y'=3$

Seleccióna
EJEMPLOS EJERCICIOS

Fig. 21.-Ventana para opción de conceptos de funciones lineales de Learn Math



- ✓ Ventana de conceptos básicos de una función polinómica y su derivada.

Definiciones

-Función polinomial: es una función en que $f(x)$ es un polinomio en x . Una función polinomial de grado n es escrita como $f(x) = ax^n + bx^{n-1} + \dots$. Las funciones polinomiales están definidas y son continuas en todos los números reales.

-Variable: Es una propiedad o un atributo que puede tomar uno o varios valores dados por los elementos de un conjunto.

-Derivada: Una derivada de una función $f(x)$ en un punto x_0 se obtiene comparando el

$f(x) = ax^n + bx^{n-1} + \dots$ $f'(x) = nax^{n-1} + (n-1)bx^{(n-1)-1} + \dots$

Gráfica de la Función y su Derivada

$y = x^3$

$y' = 3x^2$

Sección de navegación: Selección, EJEMPLOS, EJERCICIOS, Home, Next.

Fig. 22.-Ventana para opción de conceptos de funciones polinomiales de Learn Math

- ✓ Ventana de conceptos básicos de una función logarítmica y su derivada.

Definiciones

-Función logarítmica: Una función logarítmica es aquella que genéricamente se expresa como $f(x) = \log_a x$, siendo a la base de esta función, que ha de ser positiva y distinta de 1. La función logarítmica es la inversa de la función exponencial.

-Variable: Es una propiedad o un atributo que puede tomar uno o varios valores dados por los elementos de un conjunto.

-Derivada: Una derivada de una función $f(x)$

$f(x) = a \ln(x)$ $f'(x) = a/x$

Gráfica de la Función y su Derivada

$y = \ln(x)$

$y' = 1/x$

Sección de navegación: Selección, EJEMPLOS, EJERCICIOS, Home, Next.

Fig. 23.-Ventana para opción de conceptos de funciones logarítmicas de Learn Math



- ✓ Ventana de conceptos básicos de una función logarítmica y su derivada.

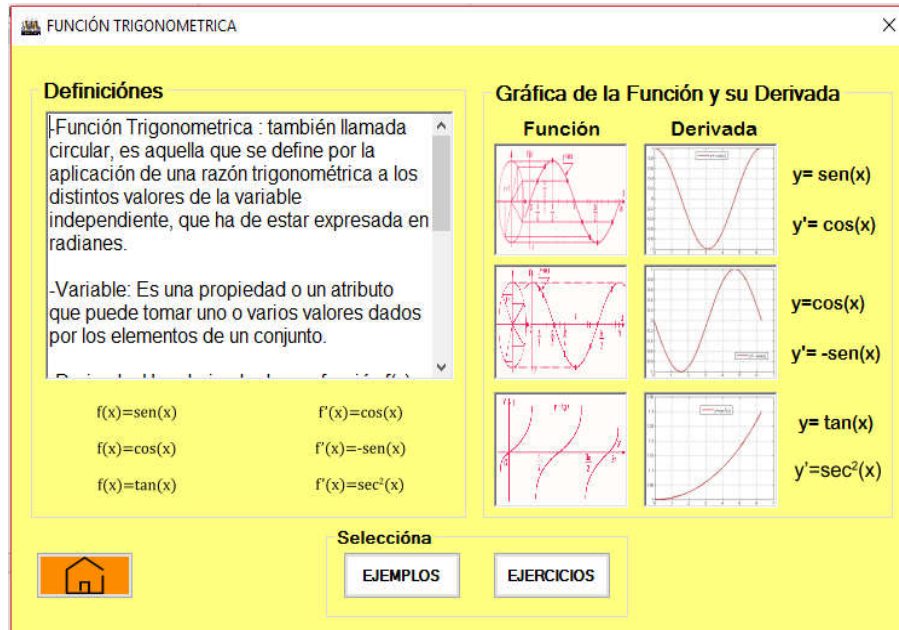


Fig. 24.-Ventana para opción de conceptos de funciones trigonométricas de Learn Math

Los conceptos en este apartado del programa son muy sencillos, esto con la finalidad de tener un concepto de referencia, sin embargo la información se debe completar con los libros.

El objetivo primordial de estas ventanas de conceptos es dar una idea general sencilla de la función y derivada básica. No se busca sustituir a un libro o a un artículo donde puede venir información más completa, El software didáctico solo es una herramienta que se podría utilizar para mejorar la clase el cual podría ser utilizado cuando el profesor lo requiera.

3.2.4 Elaboración de ventanas de presentación y menú de inicio del Learn Math.

Para el inicio del programa Learn Math se plantearon varios nombres para seleccionar solo uno el cual fue LEARN MATH que significa “Aprender Matemáticas”, ya seleccionado el



nombre, se procedió a realizar el diseño del logo el cual fue realizado en Photoshop , por lo que el diseño es propiedad del autor de esta tesis.

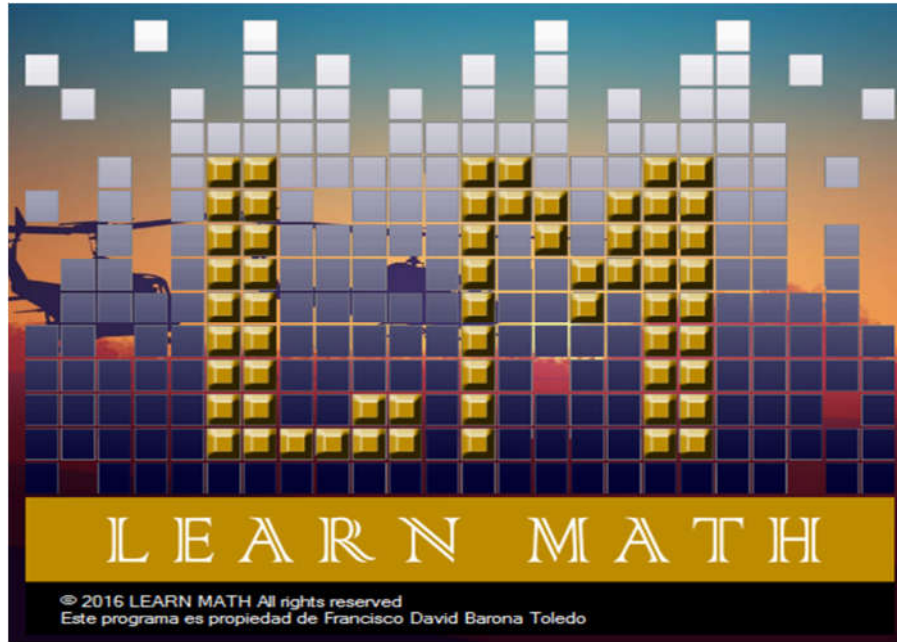


Fig. 25.-Ventana de inicio del programa Learn Math

También se realizó un ícono con esta imagen para que todas las ventanas tuvieran el icono del programa en la parte superior izquierda.

Ya con el diseño del logo del programa se pasó a programar la entrada en el programa de Visual Studio 2012, el cual es una de las herramientas más básicas para programación para la creación de aplicaciones para computadoras en ambiente de Windows.

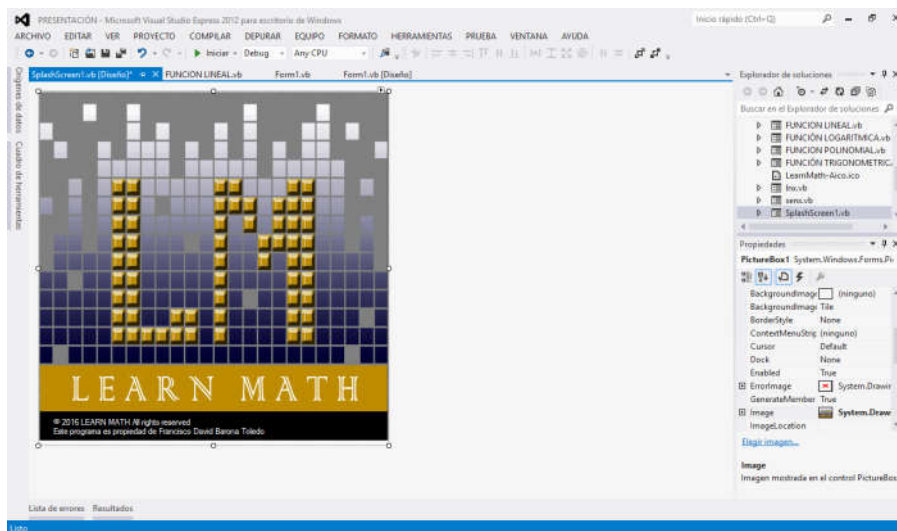


Fig. 26.-Visualización de ventana de inicio en Visual Studio.



Finalmente se procedió a realizar un menú de inicio para poder acceder a las aplicaciones antes mencionadas.

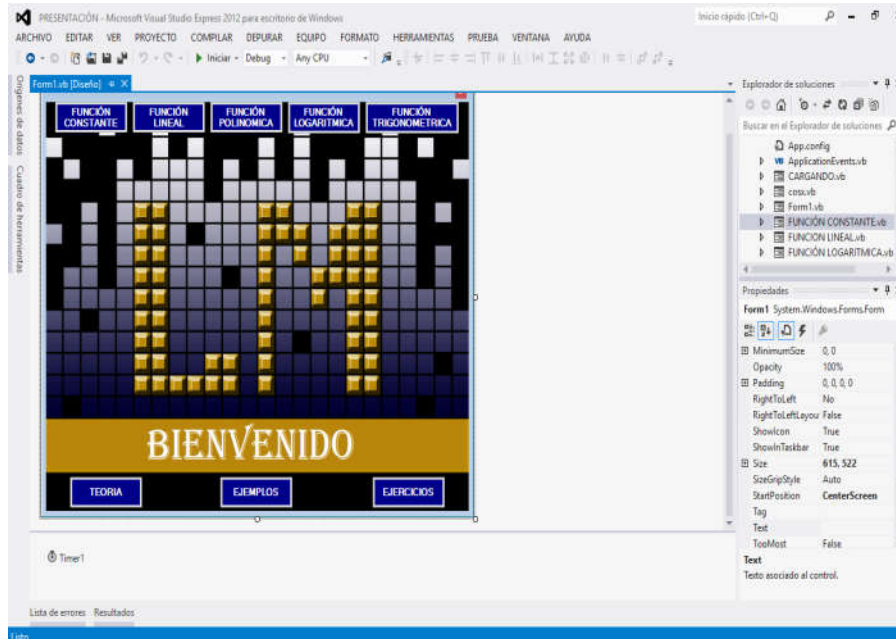


Fig. 27.-Visualizacion de ventana para menú de inicio.

3.3 Código de programación del material didáctico (Learn Math)

3.3.1 Códigos de programación para los ejemplos de Learn Math.

Los ejemplos están Programados En Simulink de Matlab, este programa es parecido a visual Studio sin embargo en Visual Studio no se pueden graficar las operaciones dadas en el código. Por lo que fue necesario usar este programa precisamente para graficar las operaciones integradas en el código y hacer un material más gráfico y demostrativo.



- ▢ Códigos de programación de los ejemplos de funciones constantes y su derivada en Learn Math.

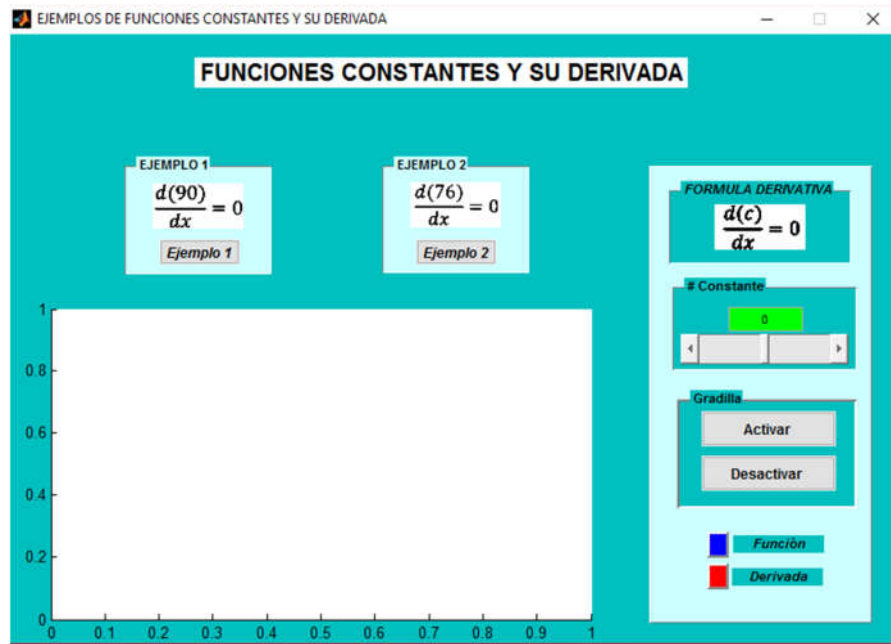


Fig. 28.-Ventana de Ejemplos de funciones constantes.

- Código de programación en botón: **Ejemplo 1**

```
function EJ1_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to EJ1 (see VARARGIN)
% Choose default command line output for EJ1
handles.output = hObject;
imag1=imread('C:\Program Files (x86)\Learn Math\dcons\d90.jpg');
subplot(handles.axes1), imshow(imag1);
imag4=imread('C:\Program Files (x86)\Learn Math\dcons\d76.jpg');
subplot(handles.axes2), imshow(imag4);
imag5=imread('C:\Program Files (x86)\Learn Math\dcons\dc.jpg');
subplot(handles.axes6), imshow(imag5);
guidata(hObject, handles);
function pusE1_Callback(hObject, eventdata, handles)
global eje1
global eje2
eje1=1;
eje2=0;
axes(handles.axes5);
x=-5:5;
```

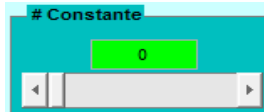


```
c=90;  
y=c*ones(size(x));  
z=0;  
y2=z*ones(size(x));  
plot(x,y,'blue',x,y2,'red','LineWidth',3)
```

- Código de programación en botón: **Ejemplo 2**

```
function pusE2_Callback(hObject, eventdata, handles)  
global eje1;  
global eje2;  
eje1=0;  
eje2=1;  
axes(handles.axes5);  
x=-5:5;  
c=76;  
y=c*ones(size(x));  
z=0;  
y2=z*ones(size(x));  
plot(x,y,'blue',x,y2,'red','LineWidth',3)
```

- Código de programación en Slider:



```
function slider1_Callback(hObject, eventdata, handles)  
va=get(handles.slider1,'value');  
set(handles.edit2,'string',va);  
axes(handles.axes5);  
x=-5:5;  
c=va;  
y=va*ones(size(x));  
z=0;  
y2=z*ones(size(x));  
plot(x,y,'blue',x,y2,'red','LineWidth',3);
```

- Código de programación en botón : **Activar**

```
function pushAc_Callback(hObject, eventdata, handles)  
set(handles.axes5,'xgrid','on','ygrid','on');
```



- Código de programación en botón : **Desactivar**

```
function pushAc_Callback(hObject, eventdata, handles)
set(handles.axes5, 'xgrid', 'on', 'ygrid', 'off');
```

- Código de programación en botón : **Función**

```
function togglebutton3_Callback(hObject, eventdata, handles)
global eje1;
global eje2;
if eje1==1;

axes(handles.axes5);

x=-5:5;
c=90;
y=c*ones(size(x));
z=0;
y2=z*ones(size(x));
plot(x,y,'blue',x,y2,'red','LineWidth', 3)
holdoff;
plot(x,y,'blue','LineWidth', 2);
legend('y=90');
clc
end
if eje2==1
axes(handles.axes5);
x=-5:5;
c=76;
y=c*ones(size(x));
z=0;
y2=z*ones(size(x));
plot(x,y,'blue',x,y2,'red','LineWidth', 3)
holdoff;
plot(x,y,'blue','LineWidth', 2);
legend('y=76');
clc
end
```

- Código de programación en botón : **Derivada**

```
function togglebutton4_Callback(hObject, eventdata, handles)
axes(handles.axes5);
```



```
x=-5:5;  
c=90;  
y=c*ones(size(x));  
z=0;  
y2=z*ones(size(x));  
plot(x,y,'blue',x,y2,'red','LineWidth',3)  
holdoff;  
plot(x,y2,'red','LineWidth',2);  
legend('y'=0');
```

- 📄 Códigos de programación de los ejemplos de funciones lineales y su derivada en Learn Math.

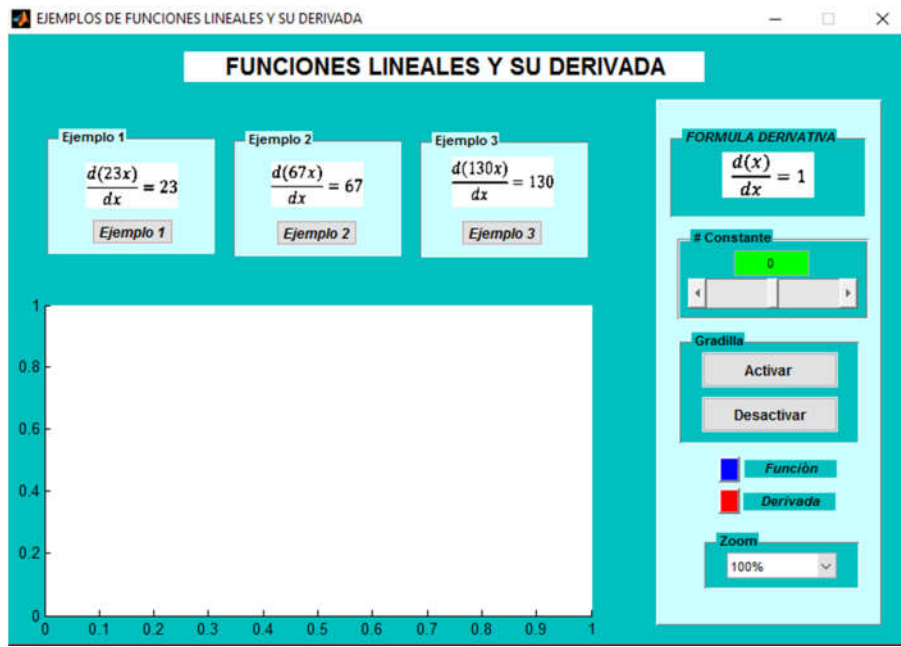


Fig. 29.-Ventana de Ejemplos de funciones lineales.

- Código de programación en botón: **Ejemplo 1**

```
function EJ2_OpeningFcn(hObject, eventdata, handles, varargin)  
% This function has no output args, see OutputFcn.  
% hObject handle to figure  
% eventdata reserved - to be defined in a future version of MATLAB  
% handles structure with handles and user data (see GUIDATA)  
% varargin command line arguments to EJ2 (see VARARGIN)  
ima1=imread('C:\Program Files (x86)\Learn Math\D1V\d23x.jpg');  
subplot(handles.axes1),imshow(ima1);  
ima4=imread('C:\Program Files (x86)\Learn Math\D1V\d67x.jpg');  
subplot(handles.axes2),imshow(ima4);  
ima5=imread('C:\Program Files (x86)\Learn Math\D1V\dx.jpg');  
subplot(handles.axes6),imshow(ima5);  
ima7=imread('C:\Program Files (x86)\Learn Math\D1V\d130x.jpg');
```



```
subplot(handles.axes9), imshow(ima7);
guidata(hObject, handles);
function pusE1_Callback(hObject, eventdata, handles)
clc
set(handles.popupmenu3, 'value', 1)
global eje3
global eje4
global eje5
eje3=1;
eje4=0;
eje5=0;
global eje6;
global eje7;
global eje8;
eje6=1;
eje7=0;
eje8=0;
global eje9;
global eje10;
global eje11;
eje9=1;
eje10=0;
eje11=0;

axes(handles.axes5);
x2=-5:5;
y2=23*ones(size(x2));
y3=23*(x2);
plot(x2,y3, 'blue', x2,y2, 'red', 'LineWidth', 3);
```

○ **Código de programación en botón:** **Ejemplo 2**

```
function pusE2_Callback(hObject, eventdata, handles)
clc
set(handles.popupmenu3, 'value', 1)
global eje3
global eje4
global eje5
eje3=0;
eje4=1;
eje5=0;
global eje6;
global eje7;
global eje8;
eje6=0;
eje7=1;
eje8=0;
global eje9;
global eje10;
global eje11;
eje9=0;
eje10=1;
eje11=0;
axes(handles.axes5);
x2=-5:5;
y2=67*ones(size(x2));
y3=67*x2;
```



```
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 3);
```

○ Código de programación en botón: **Ejemplo 3**

```
function pusE3_Callback(hObject, eventdata, handles)
set(handles.popupmenu3,'value',1)
global eje3
global eje4
global eje5
eje3=0;
eje4=0;
eje5=1;
global eje6;
global eje7;
global eje8;
eje6=0;
eje7=0;
eje8=1;
global eje9;
global eje10;
global eje11;
eje9=0;
eje10=0;
eje11=1;
axes(handles.axes5);
x2=-5:5;
y2=130*ones(size(x2));
y3=130*x2;
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 3);
```

○ Código de programación en Slider:



```
function slider1_Callback(hObject, eventdata, handles)
va=get(handles.slider1,'value');
set(handles.edit2,'string',va);
clc
axes(handles.axes5);
x2=-5:5;
y2=va*ones(size(x2));
y3=va*(x2);
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 3);
```

○ Código de programación en botón: **Activar**

```
function pushAc_Callback(hObject, eventdata, handles)
set(handles.axes5,'xgrid','on','ygrid','on');
```



- Código de programación en botón:

```
function dsg_Callback(hObject, eventdata, handles)
set(handles.axes5,'xgrid','off','ygrid','off');
```

- Código de programación en botón:

```
function togglebutton3_Callback(hObject, eventdata, handles)
global eje3;
global eje4;
global eje5;

if eje3==1;

axes(handles.axes5);
x2=-5:5;
y2=23*ones(size(x2));
y3=23*(x2);
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 3);
holdoff;
plot(x2,y3,'LineWidth', 3);
legend('y=23x');
clc
end
if eje4==1;
axes(handles.axes5);
x2=-5:5;
y2=67*ones(size(x2));
y3=67*x2;
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 3);
holdoff;
plot(x2,y3,'LineWidth', 3);
legend('y=67x');
clc
end
if eje5==1;

axes(handles.axes5);
x2=-5:5;
y2=130*ones(size(x2));
y3=130*x2;
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 3);
holdoff;
plot(x2,y3,'LineWidth', 3);
legend('y=130x');
clc
end
```



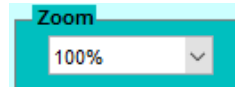
○ Código de programación en botón :  

```
function togglebutton2_Callback(hObject, eventdata, handles)
global eje6;
global eje7;
global eje8;

if eje6==1;

axes(handles.axes5);
x2=-5:5;
y2=23*ones(size(x2));
y3=23*(x2);
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 3);
holdoff;
plot(x2,y2,'red','LineWidth', 3);
legend('y'=23');
clc
end
if eje7==1;
axes(handles.axes5);
x2=-5:5;
y2=67*ones(size(x2));
y3=67*x2;
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 3);
holdoff;
plot(x2,y2,'red','LineWidth', 3);
legend('y'=67');
clc
end
if eje8==1;
axes(handles.axes5);
x2=-5:5;
y2=130*ones(size(x2));
y3=130*x2;
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 3);
holdoff;
plot(x2,y2,'red','LineWidth', 3);
legend('y'=130');
clc
end
```

○ Código de programación en pop-up Menu :



```
function popupmenu3_Callback(hObject, eventdata, handles)
dec=get(handles.popupmenu3,'value');
global eje9;
global eje10;
global eje11;
if eje9==1;
if dec==1
axis([-5 5 -150 150]*1)
end
if dec==2
```




```
axis([-5 5 -150 150]*0.75)
end
if dec==3
axis([-5 5 -150 150]*0.5)
end
if dec==4
axis([-5 5 -150 150]*0.25)
end
end
clc
if eje10==1;
if dec==2
axis([-5 5 -400 400]*0.75)
end
if dec==3
axis([-5 5 -400 400]*0.5)
end
if dec==4
axis([-5 5 -400 400]*0.25)
end
end
clc
if eje11==1;
if dec==2
axis([-5 5 -800 800]*0.75)
end
if dec==3
axis([-5 5 -800 800]*0.5)
end
if dec==4
axis([-5 5 -800 800]*0.25)
end
end
end
end
clc
```

📄 Códigos de programación de los ejemplos de funciones polinomiales y su derivada en Learn Math.

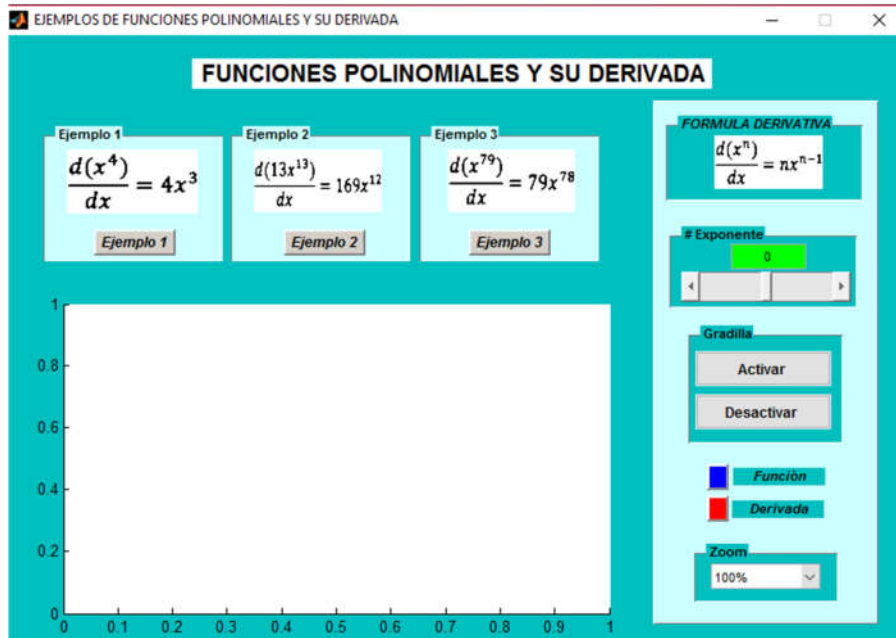


Fig. 30.-Ventana de Ejemplos de funciones polinomiales.



○ Código de programación en botón: **Ejemplo 1**

```
function EJ3_OpeningFcn(hObject, eventdata, handles, varargin)
ima1=imread('C:\Program Files (x86)\Learn Math\DEx\dx4.jpg');
subplot(handles.axes1),imshow(ima1);
ima4=imread('C:\Program Files (x86)\Learn Math\Dex\dx13.jpg');
subplot(handles.axes2),imshow(ima4);
ima5=imread('C:\Program Files (x86)\Learn Math\Dex\dxn.jpg');
subplot(handles.axes6),imshow(ima5);
ima7=imread('C:\Program Files (x86)\Learn Math\Dex\dx79.jpg');
subplot(handles.axes9),imshow(ima7);
guidata(hObject, handles);
```

```
function pusE1_Callback(hObject, eventdata, handles)
clc
set(handles.Zoom,'value',1)
global eje12
global eje13
global eje14
eje12=1;
eje13=0;
eje14=0;
global eje15
global eje16
global eje17
eje15=1;
eje16=0;
eje17=0;
global eje18
global eje19
eje18=1;
eje19=0;
axes(handles.axes5);
x2=-5:5;
y2=4*(x2.^3);
y3=(x2).^4;
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```

○ Código de programación en botón: **Ejemplo 2**

```
function pusE2_Callback(hObject, eventdata, handles)
set(handles.Zoom,'value',1)
global eje12
global eje13
global eje14
eje12=0;
eje13=1;
eje14=0;
global eje15
global eje16
global eje17
```

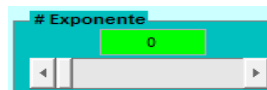


```
eje15=0;
eje16=1;
eje17=0;
global eje18
global eje19
eje18=0;
eje19=1;
axes(handles.axes5);
x2=-5:5;
y2=169*(x2.^12);
y3=(13*(x2.^13));
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```

- Código de programación en botón: **Ejemplo 3**

```
function pusE3_Callback(hObject, eventdata, handles)
clc
set(handles.Zoom,'value',1)
global eje12
global eje13
global eje14
eje12=0;
eje13=0;
eje14=1;
global eje15
global eje16
global eje17
eje15=0;
eje16=0;
eje17=1;
global eje18
global eje19
eje18=0;
eje19=0;
axes(handles.axes5);
x2=-5:5;
y2=79*(x2.^78);
plot(x2,y2,'red','LineWidth', 1);
y3=(x2).^79;
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```

- Código de programación en Slider:



```
function slider4_Callback(hObject, eventdata, handles)
va=get(handles.slider4,'value')
set(handles.edit2,'string',va)
axes(handles.axes5);
x2=-5:5;
a=va-1;
y2=va*(x2.^a);
y3=(x2).^va;
```



```
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```

- Código de programación en botón:

Activar

```
function pushAc_Callback(hObject, eventdata, handles)  
set(handles.axes5,'xgrid','on','ygrid','on');
```

- Código de programación en botón:

Desactivar

```
function dsq_Callback(hObject, eventdata, handles)  
set(handles.axes5,'xgrid','off','ygrid','off');
```

- Código de programación en botón :

Función

```
function togglebutton3_Callback(hObject, eventdata, handles)  
clc  
global eje12;  
global eje13;  
global eje14;  
if eje12==1;  
axes(handles.axes5);  
x2=-5:5;  
y2=4*(x2.^3);  
y3=(x2).^4;  
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);  
holdoff;  
plot(x2,y3,'LineWidth', 2);  
legend('y=x^4');  
clc  
end  
if eje13==1;  
axes(handles.axes5);  
x2=-5:5;  
y2=169*(x2.^12);  
y3=(13*(x2.^13));  
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);  
holdoff;  
plot(x2,y3,'LineWidth', 2);  
legend('y=13x^1^3');  
clc  
end  
if eje14==1;  
axes(handles.axes5);  
x2=-5:5;  
y2=79*(x2.^78);  
plot(x2,y2,'red','LineWidth', 1);  
y3=(x2).^79;
```

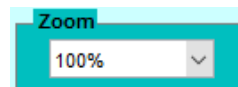


```
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
holdoff;
plot(x2,y3,'LineWidth', 2);
legend('y=x^7^9');
clc
end
```

- Código de programación en botón : 

```
function togglebutton2_Callback(hObject, eventdata, handles)
clc
global eje15
global eje16
global eje17
if eje15==1;
axes(handles.axes5);
x2=-5:5;
y2=4*(x2.^3);
y3=(x2).^4;
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
holdoff;
plot(x2,y2,'red','LineWidth', 2);
legend('y'=4x^3');
end
clc
if eje16==1;
axes(handles.axes5);
x2=-5:5;
y2=169*(x2.^12);
y3=(13*(x2.^13));
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
holdoff;
plot(x2,y2,'red','LineWidth', 2);
legend('y'=169x^1^2');
end
clc
if eje17==1;
axes(handles.axes5);
x2=-5:5;
y2=79*(x2.^78);
plot(x2,y2,'red','LineWidth', 1);
y3=(x2).^79;
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
holdoff;
plot(x2,y2,'red','LineWidth', 2);
legend('y'=79x^7^8');
end
clc
```

- Código de programación en pop-up Menu :



```
function Zoom_Callback(hObject, eventdata, handles)
dec=get(handles.Zoom,'value');
global eje18
```



```
global eje19
if eje18;
if dec==1
axis([-5 5 -600 600]*1)
end

if dec==2
axis([-5 5 -600 600]*0.75)
end
if dec==3
axis([-5 5 -600 600]*0.5)
end
if dec==4
axis([-5 5 -600 600]*0.25)
end
end
clc
if eje19;
if dec==1
axis([-5 5 -2*(10^10) 4*(10^10)]*1)
end
if dec==2
axis([-5 5 -2*(10^10) 4*(10^10)]*0.75)
end
if dec==3
axis([-5 5 -2*(10^10) 4*(10^10)]*0.5)
end
if dec==4
axis([-5 5 -2*(10^10) 4*(10^10)]*0.25)
end
end
clc
```

📄 Códigos de programación de los ejemplos de funciones logarítmicas y su derivada en Learn Math.

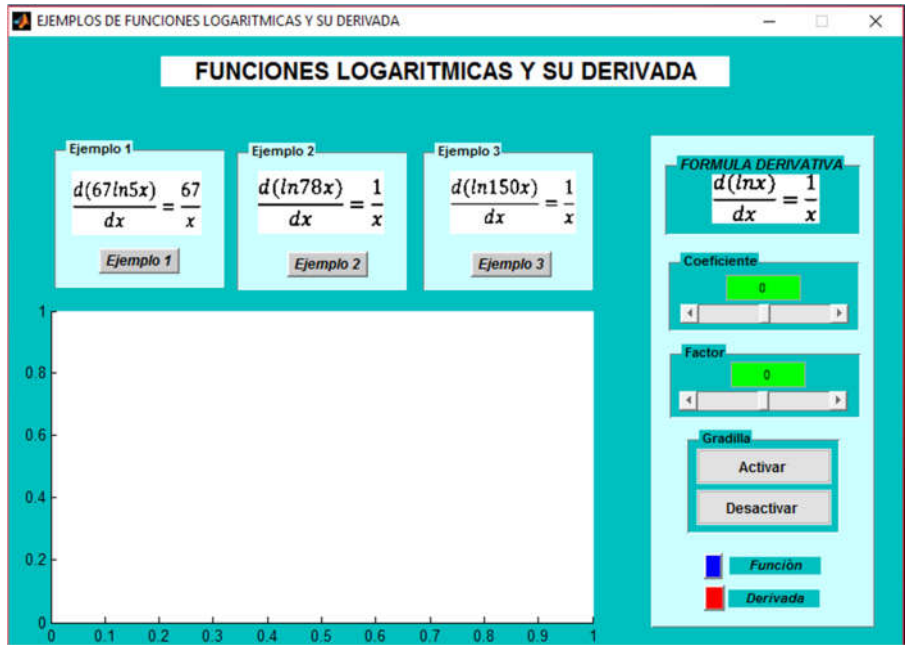


Fig.31.-Ventana de Ejemplos de funciones logarítmicas.



○ Código de programación en botón: **Ejemplo 1**

```
function EJ4_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
ima1=imread('C:\Program Files (x86)\Learn Math\DLg\d67ln5x.jpg');
subplot(handles.axes1),imshow(ima1);
ima4=imread('C:\Program Files (x86)\Learn Math\DLg\dln78x.jpg');
subplot(handles.axes2),imshow(ima4);
ima5=imread('C:\Program Files (x86)\Learn Math\DLg\dlnx.jpg');
subplot(handles.axes6),imshow(ima5);
ima7=imread('C:\Program Files (x86)\Learn Math\DLg\dln150x.jpg');
subplot(handles.axes9),imshow(ima7);
guidata(hObject, handles);
```

```
function pusE1_Callback(hObject, eventdata, handles)
global eje20
global eje21
global eje22
eje20=1;
eje21=0;
eje22=0;
global eje23
global eje24
global eje25
eje23=1;
eje24=0;
eje25=0;

axes(handles.axes5);
x2=1:5;
y2=(67./x2)
y3=67*(log(5*x2))
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```

○ Código de programación en botón: **Ejemplo 2**

```
function pusE2_Callback(hObject, eventdata, handles)
global eje20
global eje21
global eje22
eje20=0;
eje21=1;
eje22=0;
global eje23
global eje24
global eje25
eje23=0;
eje24=1;
eje25=0;
axes(handles.axes5);
x2=1:5;
```

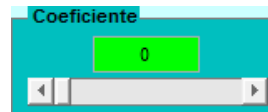


```
y2=(1./x2)
y3=log(78*x2);
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```

○ Código de programación en botón: **Ejemplo 3**

```
function pusE3_Callback(hObject, eventdata, handles)
global eje20
global eje21
global eje22
eje20=0;
eje21=0;
eje22=1;
global eje23
global eje24
global eje25
eje23=0;
eje24=0;
eje25=1;
axes(handles.axes5);
x2=1:5;
y2=(1./x2)
y3=(log(150*x2));
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```

○ Código de programación en Slider superior:



```
function slider9_Callback(hObject, eventdata, handles)
va=get(handles.slider9,'value')
set(handles.edit2,'string',va)
axes(handles.axes5);
x2=1:5;
y2=(va./x2);
y3=va*(log(x2));
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```

○ Código de programación en Slider inferior:



```
function slider10_Callback(hObject, eventdata, handles)
vi=get(handles.slider10,'value')
set(handles.edit3,'string',vi)
```




```
axes(handles.axes5);  
x2=1:5;  
y2=(1./x2);  
y3=log(vi*x2);  
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```

○ Código de programación en botón:

Activar

```
function pushAc_Callback(hObject, eventdata, handles)  
set(handles.axes5,'xgrid','on','ygrid','on');
```

○ Código de programación en botón:

Desactivar

```
function dsj_Callback(hObject, eventdata, handles)  
set(handles.axes5,'xgrid','off','ygrid','off');
```

○ Código de programación en botón :

Función

```
function togglebutton3_Callback(hObject, eventdata, handles)  
clc  
global eje20  
global eje21  
global eje22  
if eje20==1;  
axes(handles.axes5);  
x2=1:5;  
y2=(67./x2)  
y3=67*(log(5*x2))  
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);  
holdoff;  
plot(x2,y3,'LineWidth', 2);  
legend('y=67ln5x');  
clc  
end  
if eje21==1;  
axes(handles.axes5);  
x2=1:5;  
y2=(1./x2)  
y3=log(78*x2);
```



```
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
holdoff;
plot(x2,y3,'LineWidth', 2);
legend('y=ln78x');
clc
end
if eje22==1;
axes(handles.axes5);
x2=1:5;
y2=(1./x2)
y3=(log(150*x2));
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
holdoff;
plot(x2,y3,'LineWidth', 2);
legend('y=ln150x');
clc
end
```

○ Código de programación en botón : 

```
function togglebutton2_Callback(hObject, eventdata, handles)
global eje23
global eje24
global eje25
if eje23==1;
axes(handles.axes5);
x2=1:5;
y2=(67./x2);
y3=67*(log(5*x2))
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
holdoff;
plot(x2,y2,'red','LineWidth', 2);
legend('y''=67/x');
clc
end
if eje24==1;
axes(handles.axes5);
x2=1:5;
y2=(1./x2);
y3=log(78*x2);
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
holdoff;
plot(x2,y2,'red','LineWidth', 2);
legend('y''=1/x');
clc
end
if eje25==1;
axes(handles.axes5);
x2=1:5;
y2=(1./x2);
y3=(log(150*x2));
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
holdoff;
plot(x2,y2,'red','LineWidth', 2);
legend('y''=1/x');
clc
end
```



- 📄 Códigos de programación de los ejemplos de funciones trigonométricas y su derivada en Learn Math.

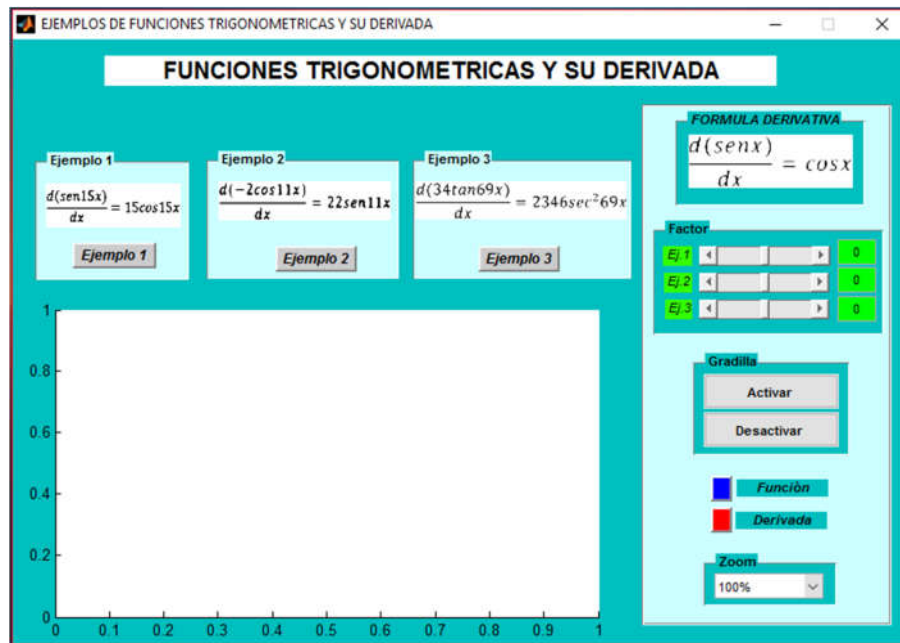


Fig.32.-Ventana de Ejemplos de funciones Trigonómicas.

- Código de programación en botón: **Ejemplo 1**

```
function EJ5_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
ima1=imread('C:\Program Files (x86)\Learn Math\Dtri\dsen15x.jpg');
subplot(handles.axes1), imshow(ima1);
ima4=imread('C:\Program Files (x86)\Learn Math\Dtri\d2cos11x.jpg');
subplot(handles.axes2), imshow(ima4);
ima5=imread('C:\Program Files (x86)\Learn Math\Dtri\dsenx.jpg');
subplot(handles.axes6), imshow(ima5);
ima7=imread('C:\Program Files (x86)\Learn Math\Dtri\d34tan69x.jpg');
subplot(handles.axes9), imshow(ima7);
guidata(hObject, handles);
```

```
function pusE1_Callback(hObject, eventdata, handles)
set(handles.Zoom, 'value', 1)
global eje26
global eje27
global eje28
eje26=1;
eje27=0;
eje28=0;
```



```
global eje29
global eje30
global eje31
eje29=1;
eje30=0;
eje31=0;
global eje32
global eje33
global eje34
eje32=1;
eje33=0;
eje34=0;
axes(handles.axes5);
x2=linspace(0,1);
y2=15*cos(15*x2);
y3=sin(15*x2);
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```

- Código de programación en botón: **Ejemplo 2**

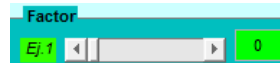
```
function pusE2_Callback(hObject, eventdata, handles)
set(handles.Zoom,'value',1)
global eje26
global eje27
global eje28
eje26=0;
eje27=1;
eje28=0;
global eje29
global eje30
global eje31
eje29=0;
eje30=1;
eje31=0;
global eje32
global eje33
global eje34
eje32=0;
eje33=1;
eje34=0;
ima10=imread('F:\MT\Dtri\dcosx.jpg');
subplot(handles.axes6),imshow(ima10);
axes(handles.axes5);
x2=linspace(0,1);
y2=22*(sin(11*x2));
y3=-2*cos(11*x2);
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```



○ Código de programación en botón: **Ejemplo 3**

```
function pusE3_Callback(hObject, eventdata, handles)
set(handles.Zoom, 'value', 1)
global eje26
global eje27
global eje28
eje26=0;
eje27=0;
eje28=1;
global eje29
global eje30
global eje31
eje29=0;
eje30=0;
eje31=1;
global eje32
global eje33
global eje34
eje32=0;
eje33=0;
eje34=1;
ima1=imread('F:\MT\Dtri\dtanx.jpg');
subplot(handles.axes6), imshow(ima1);
axes(handles.axes5);
x2=linspace(0,0.5);
y2=2346*(sec(69*x2).^2);
y3=34*tan(69*x2);
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```

○ Código de programación en Slider Primario:



```
function slider4_Callback(hObject, eventdata, handles)
va=get(handles.slider4, 'value')
set(handles.edit2, 'string', va)
axes(handles.axes5);
x2=linspace(0,1);
y2=va*cos(va*x2);
y3=sin(va*x2)
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```

○ Código de programación en Slider de secundaria:



```
function slider5_Callback(hObject, eventdata, handles)
vi=get(handles.slider5, 'value')
set(handles.edit3, 'string', vi)
axes(handles.axes5);
```



```
x2=linspace(0,1);  
y2=(-vi*(sin(vi*x2)));  
y3=cos(vi*x2)  
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```

- **Código de programación en Slider de enmedio:**

```
function slider6_Callback(hObject, eventdata, handles)  
ve=get(handles.slider6,'value')  
set(handles.edit4,'string',ve)  
axes(handles.axes5);  
x2=linspace(0,1);  
y2=ve*(sec(ve*x2).^2);  
y3=tan(ve*x2)  
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```

- **Código de programación en botón:**

```
function pushAc_Callback(hObject, eventdata, handles)  
set(handles.axes5,'xgrid','on','ygrid','on');
```

- **Código de programación en botón:**

```
function dsg_Callback(hObject, eventdata, handles)  
set(handles.axes5,'xgrid','off','ygrid','off');
```

- **Código de programación en botón:**

```
function togglebutton3_Callback(hObject, eventdata, handles)  
global eje26  
global eje27  
global eje28  
if eje26==1;  
axes(handles.axes5);  
x2=linspace(0,1);  
y2=15*cos(15*x2);  
y3=sin(15*x2)  
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
```



```
holdoff;
plot(x2,y3,'LineWidth', 2);
legend('y=seno(15x)');
clc
end
if eje27==1;
axes(handles.axes5);
x2=linspace(0,1);
y2=22*(sin(11*x2));
y3=-2*cos(11*x2)
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
holdoff;
plot(x2,y3,'LineWidth', 2);
legend('y=-2cos(11x)');
clc
end
if eje28==1;
axes(handles.axes5);
x2=linspace(0,0.5);
y2=2346*(tan(69*x2).^2);
y3=34*tan(69*x2)
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
holdoff;
plot(x2,y3,'LineWidth', 2);
legend('y=34tan(69x)');
clc
end
```

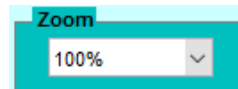
- Código de programación en botón : 

```
function togglebutton2_Callback(hObject, eventdata, handles)
global eje29
global eje30
global eje31
if eje29==1;
axes(handles.axes5);
x2=linspace(0,1);
y2=15*cos(15*x2);
y3=sin(15*x2)
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
holdoff;
plot(x2,y2,'red','LineWidth', 2);
legend('y'=15cos(15x)');
clc
end
if eje30==1;
axes(handles.axes5);
x2=linspace(0,1);
y2=22*(sin(11*x2));
y3=-2*cos(11*x2)
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
holdoff;
plot(x2,y2,'red','LineWidth', 2);
legend('y'=22sen(11x)');
clc
end
```



```
if eje31==1;
axes(handles.axes5);
x2=linspace(0,0.5);
y2=2346*(sec(69*x2).^2);
y3=34*tan(69*x2)
plot(x2,y3,'blue',x2,y2,'red','LineWidth', 2);
holdoff;
plot(x2,y2,'red','LineWidth', 2);
legend('y''=234sec^2(69x)');
clc
end
```

○ Código de programación en pop-up Menu :



```
function Zoom_Callback(hObject, eventdata, handles)
dec=get(handles.Zoom,'value');
global eje32
global eje33
global eje34
if eje32==1;
if dec==1
axis([0 1 -15 15]*1)
end

if dec==2
axis([0 1 -15 15]*0.75)
end
if dec==3
axis([0 1 -15 15]*0.5)
end
if dec==4
axis([0 1 -15 15]*0.25)
end
end
clc
if eje33==1;
if dec==1
axis([0 1 -30 30]*1)
end
if dec==2
axis([0 1 -30 30]*0.75)
end
if dec==3
axis([0 1 -30 30]*0.5)
end
if dec==4
axis([0 1 -30 30]*0.25)
end
end
clc
if eje34==1;
if dec==1
axis([0 1 -5 20*(10^4)]*1)
end
```




```
if dec==2  
axis([0 1 -5 20*(10^4)]*0.75)  
end  
if dec==3  
axis([0 1 -5 20*(10^4)]*0.5)  
end  
if dec==4  
axis([0 1 -5 20*(10^4)]*0.25)  
end  
end
```

3.3.2 Códigos de programación para los ejercicios de Learn Math.

Las ventanas para ejercicios propuestos en Learn Math, fueron realizados en Visual Studio 2012 ya que es una herramienta de programación básica en la cual se puede utilizar Visual Studio para crear aplicaciones para la Tienda de Windows, aplicaciones de escritorio, aplicaciones móviles, aplicaciones web ASP.NET y servicios web XML donde se puede escribir código en Visual Basic, Visual C#, Visual C++, Visual F# y JavaScript, y crear soluciones con lenguajes mixtos más fácilmente. Además de ser una herramienta utilizada durante la formación del Ingeniero Químico como una herramienta para hacer programas para simulación para crear botones, casillas y pocos cálculos matemáticos además de ser un programa cuya estética mejora a comparación del simulink de Matlab. También el nivel de inglés necesario para manipular el programa es bastante básico y a su vez tiene opción de autocompletado de palabras para poder guiarte de una manera más fácil.

- 📄 Códigos de programación de los ejercicios de funciones constantes y su derivada en Learn Math.

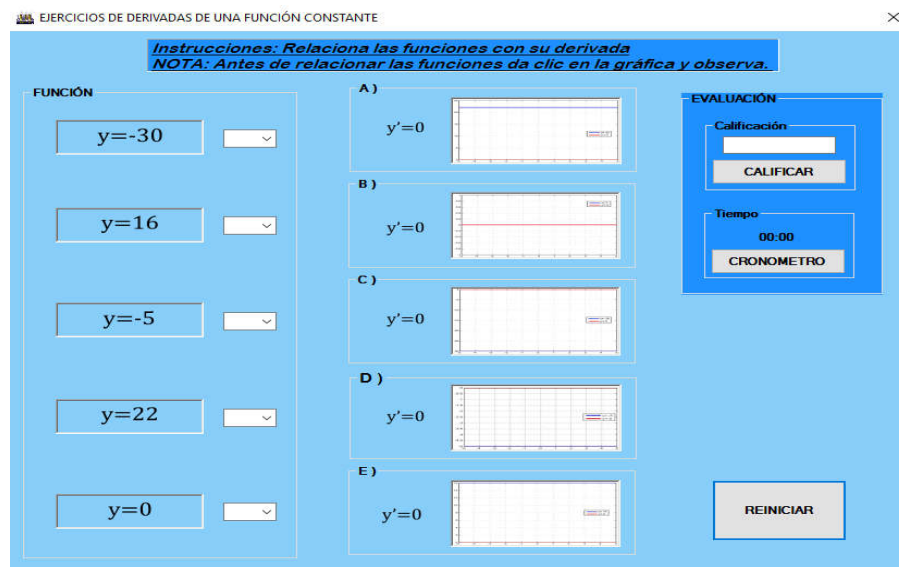


Fig.33.-Ventana de Ejercicios de funciones constantes.



- Código de programación en botón : CALIFICAR

```
PublicClassForm1
Dim seg, min AsInteger

PrivateSub CALIFICAR_Click(sender AsObject, e AsEventArgs) Handles CALIFICAR.Click
Dim SUMS AsInteger = 0
IfVal(ComboBox1.SelectedIndex) = 2 Then
    ComboBox1.BackColor = Color.Gold
    GroupBox4.BackColor = Color.Gold
    ComboBox1.ForeColor = Color.Black
    SUMS = SUMS + 2
Else
    ComboBox1.BackColor = Color.Red
    GroupBox4.BackColor = Color.Red
    ComboBox1.ForeColor = Color.White
EndIf

IfVal(ComboBox2.SelectedIndex) = 4 Then
    ComboBox2.BackColor = Color.Gold
    GroupBox6.BackColor = Color.Gold
    ComboBox2.ForeColor = Color.Black
    SUMS = SUMS + 2
Else
    ComboBox2.BackColor = Color.Red
    GroupBox6.BackColor = Color.Red
    ComboBox2.ForeColor = Color.White
EndIf

IfVal(ComboBox3.SelectedIndex) = 3 Then
    ComboBox3.BackColor = Color.Gold
    GroupBox5.BackColor = Color.Gold
    ComboBox3.ForeColor = Color.Black
    SUMS = SUMS + 2
Else
    ComboBox3.BackColor = Color.Red
    GroupBox5.BackColor = Color.Red
    ComboBox3.ForeColor = Color.White
EndIf

IfVal(ComboBox4.SelectedIndex) = 0 Then
    ComboBox4.BackColor = Color.Gold
    GroupBox2.BackColor = Color.Gold
    ComboBox4.ForeColor = Color.Black
    SUMS = SUMS + 2
Else
    ComboBox4.BackColor = Color.Red
    GroupBox2.BackColor = Color.Red
    ComboBox4.ForeColor = Color.White
EndIf

IfVal(ComboBox5.SelectedIndex) = 1 Then
    ComboBox5.BackColor = Color.Gold
    GroupBox3.BackColor = Color.Gold
    ComboBox5.ForeColor = Color.Black
    SUMS = SUMS + 2
Else
    ComboBox5.BackColor = Color.Red
    GroupBox3.BackColor = Color.Red
```



```
        ComboBox5.ForeColor = Color.White
EndIf
    TextBox1.Text = SUMS
Timer1.Stop()
SelectCase SUMS
Case 0 To 5
    PictureBox17.Visible = True
REPRO.Show()
Case 6 To 10
    PictureBox16.Visible = True
GANASTE.Show()
EndSelect
EndSub
```

- **Código de programación en botón :** 

```
PrivateSub Button2_Click(sender AsObject, e AsEventArgs) Handles Button2.Click
    seg = 0
    min = 0
    Label7.Text = "00:00"
    Timer1.Start()
EndSub
```

- **Código de programación en Timer :** 

```
PrivateSub Timer1_Tick(sender AsObject, e AsEventArgs) Handles Timer1.Tick
    seg += 1
    If seg = 60 Then
        min += 1
        seg = 0
    EndIf
    If min < 10 And seg < 10 Then
        Label7.Text = "0"& min & ":0"& seg
    ElseIf min < 10 And seg > 10 Then
        Label7.Text = "0"& min & ":"& seg
    ElseIf min > 10 And seg > 10 Then
        Label7.Text = "0"& min & ":"& seg
    ElseIf min > 10 And seg < 10 Then
        Label7.Text = min & ":"& seg
    EndIf
EndSub
```

- **Código de programación en botón :** 

```
ComboBox1.Text = Nothing : ComboBox2.Text = Nothing : ComboBox3.Text = Nothing :
ComboBox4.Text = Nothing : ComboBox5.Text = Nothing
ComboBox1.Focus()
```



```
ComboBox1.BackColor = Color.White  
ComboBox1.ForeColor = Color.Black  
ComboBox2.BackColor = Color.White  
ComboBox2.ForeColor = Color.Black  
ComboBox3.BackColor = Color.White  
ComboBox3.ForeColor = Color.Black  
ComboBox4.BackColor = Color.White  
ComboBox4.ForeColor = Color.Black  
ComboBox5.BackColor = Color.White  
ComboBox5.ForeColor = Color.Black  
ComboBox5.BackColor = Color.White  
  
GroupBox2.BackColor = Color.LightSkyBlue  
GroupBox3.BackColor = Color.LightSkyBlue  
GroupBox4.BackColor = Color.LightSkyBlue  
GroupBox5.BackColor = Color.LightSkyBlue  
GroupBox6.BackColor = Color.LightSkyBlue  
  
TextBox1.Text = Nothing  
Timer1.Stop()  
Label7.Text = "00:00"  
  
PictureBox17.Visible = False  
PictureBox16.Visible = False  
EndSub
```

📄 Códigos de programación de los ejercicios de funciones lineales y su derivada en Learn Math.

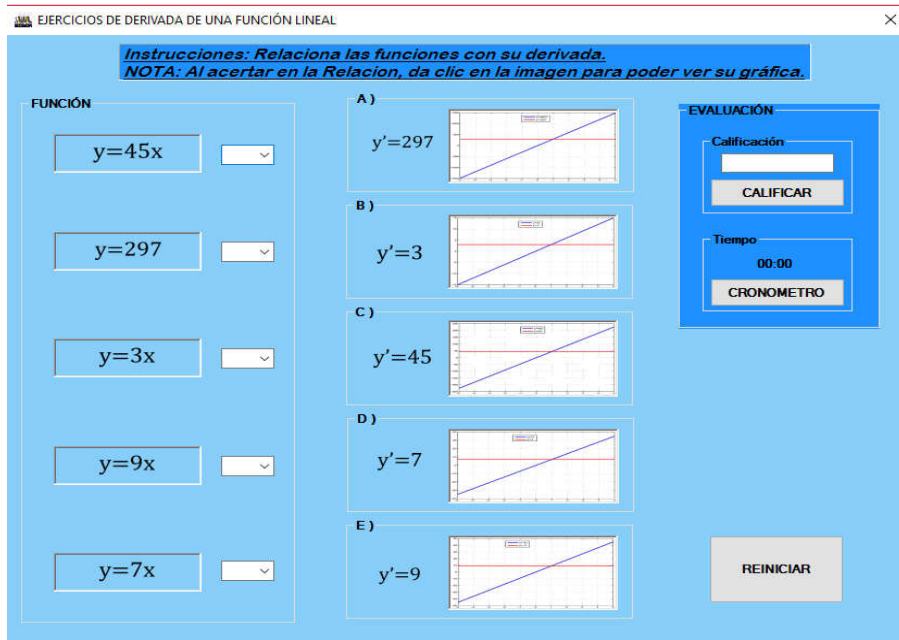


Fig.34.-Ventana de Ejercicios de funciones lineales.



- Código de programación en botón :

CALIFICAR

```
PublicClassForm1
```

```
Dim seg, min AsInteger
```

```
PrivateSub CALIFICAR_Click(sender AsObject, ByVal e AsEventArgs) Handles CALIFICAR.Click
```

```
Dim SUMS AsInteger = 0
```

```
IfVal(ComboBox1.SelectedIndex) = 2 Then  
    ComboBox1.BackColor = Color.Gold  
    GroupBox4.BackColor = Color.Gold  
    ComboBox1.ForeColor = Color.Black  
    SUMS = SUMS + 2  
    PictureBox10.Enabled = True
```

```
Else  
    ComboBox1.BackColor = Color.Red  
    GroupBox4.BackColor = Color.Red  
    ComboBox1.ForeColor = Color.White  
    PictureBox10.Enabled = False
```

```
EndIf
```

```
IfVal(ComboBox2.SelectedIndex) = 0 Then  
    ComboBox2.BackColor = Color.Gold  
    GroupBox2.BackColor = Color.Gold  
    ComboBox2.ForeColor = Color.Black  
    SUMS = SUMS + 2  
    PictureBox7.Enabled = True
```

```
Else  
    ComboBox2.BackColor = Color.Red  
    GroupBox2.BackColor = Color.Red  
    ComboBox2.ForeColor = Color.White  
    PictureBox7.Enabled = False
```

```
EndIf
```

```
IfVal(ComboBox3.SelectedIndex) = 1 Then  
    ComboBox3.BackColor = Color.Gold  
    GroupBox3.BackColor = Color.Gold  
    ComboBox3.ForeColor = Color.Black  
    SUMS = SUMS + 2  
    PictureBox8.Enabled = True
```

```
Else  
    ComboBox3.BackColor = Color.Red  
    GroupBox3.BackColor = Color.Red  
    ComboBox3.ForeColor = Color.White  
    PictureBox8.Enabled = False
```

```
EndIf
```

```
IfVal(ComboBox4.SelectedIndex) = 4 Then  
    ComboBox4.BackColor = Color.Gold  
    GroupBox6.BackColor = Color.Gold  
    ComboBox4.ForeColor = Color.Black  
    SUMS = SUMS + 2  
    PictureBox14.Enabled = True
```

```
Else  
    ComboBox4.BackColor = Color.Red  
    GroupBox6.BackColor = Color.Red  
    ComboBox4.ForeColor = Color.White  
    PictureBox14.Enabled = False
```

```
EndIf
```

```
IfVal(ComboBox5.SelectedIndex) = 3 Then
```



```
ComboBox5.BackColor = Color.Gold
GroupBox5.BackColor = Color.Gold
ComboBox5.ForeColor = Color.Black
SUMS = SUMS + 2
PictureBox12.Enabled = True

Else
    ComboBox5.BackColor = Color.Red
    GroupBox5.BackColor = Color.Red
    ComboBox5.ForeColor = Color.White
    PictureBox12.Enabled = False

EndIf
    TextBox1.Text = SUMS
Timer1.Stop()
Select SUMS
Case 0 To 5
    PictureBox17.Visible = True
REPRO.Show()
Case 6 To 10
    PictureBox16.Visible = True
GANASTE.Show()
EndSelect
EndSub
```

- Código de programación en botón : 

```
PrivateSub Button2_Click(sender AsObject, e AsEventArgs) Handles Button2.Click
    seg = 0
    min = 0
    Label7.Text = "00:00"
    Timer1.Start()
EndSub
```

- Código de programación en Timer : 

```
PrivateSub Timer1_Tick(sender AsObject, e AsEventArgs) Handles Timer1.Tick
    seg += 1
    If seg = 60 Then
        min += 1
        seg = 0
    EndIf
    If min < 10 And seg < 10 Then
        Label7.Text = "0"& min & ":"& 0"& seg
    ElseIf min < 10 And seg > 10 Then
        Label7.Text = "0"& min & ":"& "& seg
    ElseIf min > 10 And seg > 10 Then
        Label7.Text = "0"& min & ":"& "& seg
    ElseIf min > 10 And seg < 10 Then
        Label7.Text = min & ":"& "& seg
    EndIf
EndSub
```



- Código de programación en botón :



```
ComboBox1.Text = Nothing : ComboBox2.Text = Nothing : ComboBox3.Text = Nothing :  
ComboBox4.Text = Nothing : ComboBox5.Text = Nothing  
ComboBox1.Focus()
```

```
ComboBox1.BackColor = Color.White  
ComboBox1.ForeColor = Color.Black  
ComboBox2.BackColor = Color.White  
ComboBox2.ForeColor = Color.Black  
ComboBox3.BackColor = Color.White  
ComboBox3.ForeColor = Color.Black  
ComboBox4.BackColor = Color.White  
ComboBox4.ForeColor = Color.Black  
ComboBox5.BackColor = Color.White  
ComboBox5.ForeColor = Color.Black  
ComboBox5.BackColor = Color.White  
GroupBox2.BackColor = Color.LightSkyBlue  
GroupBox3.BackColor = Color.LightSkyBlue  
GroupBox4.BackColor = Color.LightSkyBlue  
GroupBox5.BackColor = Color.LightSkyBlue  
GroupBox6.BackColor = Color.LightSkyBlue  
TextBox1.Text = Nothing
```

```
Timer1.Stop()  
Label7.Text = "00:00"  
PictureBox17.Visible = False  
PictureBox16.Visible = False
```

EndSu

- 📄 Códigos de programación de los ejercicios de funciones polinomiales y su derivada en Learn Math.

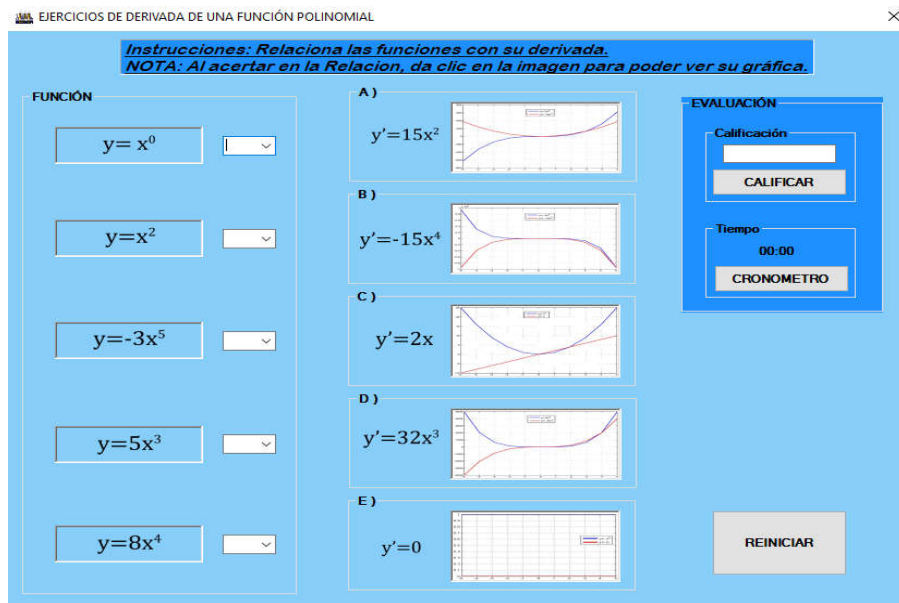


Fig.35.-Ventana de Ejercicios de funciones polinomiales.



○ Código de programación en botón :

CALIFICAR

```
PrivateSub CALIFICAR_Click(sender AsObject, e AsEventArgs) Handles CALIFICAR.Click  
Dim SUMS AsInteger = 0
```

```
    IfVal(ComboBox1.SelectedIndex) = 4 Then  
        ComboBox1.BackColor = Color.Gold  
        GroupBox6.BackColor = Color.Gold  
        ComboBox1.ForeColor = Color.Black  
        SUMS = SUMS + 2
```

```
    Else  
        ComboBox1.BackColor = Color.Red  
        GroupBox6.BackColor = Color.Red  
        ComboBox1.ForeColor = Color.White
```

```
EndIf  
    IfVal(ComboBox2.SelectedIndex) = 2 Then  
        ComboBox2.BackColor = Color.Gold  
        GroupBox4.BackColor = Color.Gold  
        ComboBox2.ForeColor = Color.Black  
        SUMS = SUMS + 2
```

```
    Else  
        ComboBox2.BackColor = Color.Red  
        GroupBox4.BackColor = Color.Red  
        ComboBox2.ForeColor = Color.White
```

```
EndIf  
    IfVal(ComboBox3.SelectedIndex) = 1 Then  
        ComboBox3.BackColor = Color.Gold  
        GroupBox3.BackColor = Color.Gold  
        ComboBox3.ForeColor = Color.Black  
        SUMS = SUMS + 2
```

```
    Else  
        ComboBox3.BackColor = Color.Red  
        GroupBox3.BackColor = Color.Red  
        ComboBox3.ForeColor = Color.White
```

```
EndIf  
    IfVal(ComboBox4.SelectedIndex) = 0 Then  
        ComboBox4.BackColor = Color.Gold  
        GroupBox2.BackColor = Color.Gold  
        ComboBox4.ForeColor = Color.Black  
        SUMS = SUMS + 2
```

```
    Else  
        ComboBox4.BackColor = Color.Red  
        GroupBox2.BackColor = Color.Red  
        ComboBox4.ForeColor = Color.White
```

```
EndIf  
    IfVal(ComboBox5.SelectedIndex) = 3 Then  
        ComboBox5.BackColor = Color.Gold  
        GroupBox5.BackColor = Color.Gold  
        ComboBox5.ForeColor = Color.Black  
        SUMS = SUMS + 2
```

```
    Else  
        ComboBox5.BackColor = Color.Red  
        GroupBox5.BackColor = Color.Red  
        ComboBox5.ForeColor = Color.White
```

```
EndIf  
    TextBox1.Text = SUMS  
Timer1.Stop()  
SelectCase SUMS  
Case 0 To 5
```




```
PictureBox17.Visible = True  
REPRO.Show()  
Case 6 To 10  
PictureBox16.Visible = True  
GANASTE.Show()  
EndSelect  
EndSub
```

- Código de programación en botón : 

```
PrivateSub Button2_Click(sender AsObject, e AsEventArgs) Handles Button2.Click  
seg = 0  
min = 0  
Label7.Text = "00:00"  
Timer1.Start()  
EndSub
```

- Código de programación en Timer : 

```
PrivateSub Timer1_Tick(sender AsObject, e AsEventArgs) Handles Timer1.Tick  
seg += 1  
If seg = 60 Then  
min += 1  
seg = 0  
EndIf  
If min < 10 And seg < 10 Then  
Label7.Text = "0"& min & ":0"& seg  
ElseIf min < 10 And seg > 10 Then  
Label7.Text = "0"& min & ":"& seg  
ElseIf min > 10 And seg > 10 Then  
Label7.Text = "0"& min & ":"& seg  
ElseIf min > 10 And seg < 10 Then  
Label7.Text = min & ":"& seg  
EndIf  
EndSub  
  
If min < 10 And seg < 10 Then  
Label7.Text = "0"& min & ":0"& seg  
ElseIf min < 10 And seg > 10 Then  
Label7.Text = "0"& min & ":"& seg  
ElseIf min > 10 And seg > 10 Then  
Label7.Text = "0"& min & ":"& seg  
ElseIf min > 10 And seg < 10 Then  
Label7.Text = min & ":"& seg  
EndIf  
EndSub
```



- Código de programación en botón :

REINICIAR

```
ComboBox1.Text = Nothing : ComboBox2.Text = Nothing : ComboBox3.Text = Nothing :  
ComboBox4.Text = Nothing : ComboBox5.Text = Nothing  
ComboBox1.Focus()  
    ComboBox1.BackColor = Color.White  
    ComboBox1.ForeColor = Color.Black  
    ComboBox2.BackColor = Color.White  
    ComboBox2.ForeColor = Color.Black  
    ComboBox3.BackColor = Color.White  
    ComboBox3.ForeColor = Color.Black  
    ComboBox4.BackColor = Color.White  
    ComboBox4.ForeColor = Color.Black  
    ComboBox5.BackColor = Color.White  
    ComboBox5.ForeColor = Color.Black  
    ComboBox5.BackColor = Color.White  
    GroupBox2.BackColor = Color.LightSkyBlue  
    GroupBox3.BackColor = Color.LightSkyBlue  
    GroupBox4.BackColor = Color.LightSkyBlue  
    GroupBox5.BackColor = Color.LightSkyBlue  
    GroupBox6.BackColor = Color.LightSkyBlue  
Timer1.Stop()  
    TextBox1.Text = Nothing  
    Label7.Text = "00:00"  
Dim SUMS As Integer = 0  
    SUMS = 0  
    PictureBox17.Visible = False  
    PictureBox16.Visible = False  
EndSub
```

- 📄 Códigos de programación de los ejercicios de funciones logarítmicas y su derivada en Learn Math.

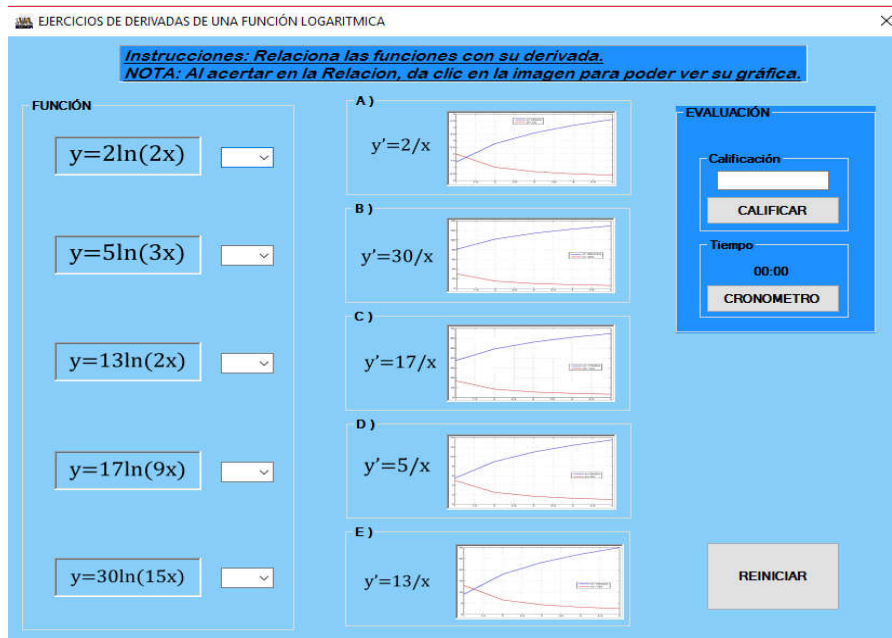


Fig.36.-Ventana de Ejercicios de funciones logarítmicas.



- Código de programación en botón :

CALIFICAR

```
PublicClassForm1
Dim seg, min AsInteger

PrivateSub CALIFICAR_Click(sender AsObject, e AsEventArgs) Handles CALIFICAR.Click
Dim SUMS AsInteger = 0
IfVal(ComboBox1.SelectedIndex) = 0 Then
    ComboBox1.BackColor = Color.Gold
    GroupBox2.BackColor = Color.Gold
    ComboBox1.ForeColor = Color.Black
    SUMS = SUMS + 2
    PictureBox7.Enabled = True
Else
    ComboBox1.BackColor = Color.Red
    GroupBox2.BackColor = Color.Red
    ComboBox1.ForeColor = Color.White
EndIf
IfVal(ComboBox2.SelectedIndex) = 3 Then
    ComboBox2.BackColor = Color.Gold
    GroupBox5.BackColor = Color.Gold
    ComboBox2.ForeColor = Color.Black
    SUMS = SUMS + 2
    PictureBox12.Enabled = True
Else
    ComboBox2.BackColor = Color.Red
    GroupBox5.BackColor = Color.Red
    ComboBox2.ForeColor = Color.White
EndIf
IfVal(ComboBox3.SelectedIndex) = 4 Then
    ComboBox3.BackColor = Color.Gold
    GroupBox6.BackColor = Color.Gold
    ComboBox3.ForeColor = Color.Black
    SUMS = SUMS + 2
    PictureBox14.Enabled = True
Else
    ComboBox3.BackColor = Color.Red
    GroupBox6.BackColor = Color.Red
    ComboBox3.ForeColor = Color.White
EndIf
IfVal(ComboBox4.SelectedIndex) = 2 Then
    ComboBox4.BackColor = Color.Gold
    GroupBox4.BackColor = Color.Gold
    ComboBox4.ForeColor = Color.Black
    SUMS = SUMS + 2
    PictureBox10.Enabled = True
Else
    ComboBox4.BackColor = Color.Red
    GroupBox4.BackColor = Color.Red
    ComboBox4.ForeColor = Color.White
EndIf
IfVal(ComboBox5.SelectedIndex) = 1 Then
    ComboBox5.BackColor = Color.Gold
    GroupBox3.BackColor = Color.Gold
    ComboBox5.ForeColor = Color.Black
    SUMS = SUMS + 2
    PictureBox8.Enabled = True
Else
```



```
ComboBox5.BackColor = Color.Red
GroupBox3.BackColor = Color.Red
ComboBox5.ForeColor = Color.White

EndIf
    TextBox1.Text = SUMS
Timer1.Stop()
SelectCase SUMS
Case 0 To 5
    PictureBox17.Visible = True
REPRO.Show()
Case 6 To 10
    PictureBox16.Visible = True
GANASTE.Show()
EndSelect
EndSub
```

- **Código de programación en botón :** 

```
PrivateSub Button2_Click(sender AsObject, e AsEventArgs) Handles Button2.Click
seg = 0
min = 0
    Label7.Text = "00:00"
Timer1.Start()EndSub
```

- **Código de programación en Timer :** 

```
PrivateSub Timer1_Tick(sender AsObject, e AsEventArgs) Handles Timer1.Tick
seg += 1
If seg = 60 Then
min += 1
seg = 0
EndIf
If min < 10 And seg < 10 Then
    Label7.Text = "0"& min & ":0"& seg
ElseIf min < 10 And seg > 10 Then
    Label7.Text = "0"& min & ":"& seg
ElseIf min > 10 And seg > 10 Then
    Label7.Text = "0"& min & ":"& seg
ElseIf min > 10 And seg < 10 Then
    Label7.Text = min & ":"& seg
EndIf
EndSub
```

- **Código de programación en botón :** 

```
ComboBox1.Text = Nothing : ComboBox2.Text = Nothing : ComboBox3.Text = Nothing :
ComboBox4.Text = Nothing : ComboBox5.Text = Nothing
```



```
ComboBox1.Focus()  
    ComboBox1.BackColor = Color.White  
    ComboBox1.ForeColor = Color.Black  
    ComboBox2.BackColor = Color.White  
    ComboBox2.ForeColor = Color.Black  
    ComboBox3.BackColor = Color.White  
    ComboBox3.ForeColor = Color.Black  
    ComboBox4.BackColor = Color.White  
    ComboBox4.ForeColor = Color.Black  
    ComboBox5.BackColor = Color.White  
    ComboBox5.ForeColor = Color.Black  
    ComboBox5.BackColor = Color.White  
    GroupBox2.BackColor = Color.LightSkyBlue  
    GroupBox3.BackColor = Color.LightSkyBlue  
    GroupBox4.BackColor = Color.LightSkyBlue  
    GroupBox5.BackColor = Color.LightSkyBlue  
    GroupBox6.BackColor = Color.LightSkyBlue  
Timer1.Stop()  
    TextBox1.Text = Nothing  
    Label7.Text = "00:00"  
Dim SUMS As Integer = 0  
    SUMS = 0  
    PictureBox17.Visible = False  
    PictureBox16.Visible = False  
EndSub
```

- 📄 Códigos de programación de los ejercicios de funciones trigonométricas y su derivada en Learn Math.

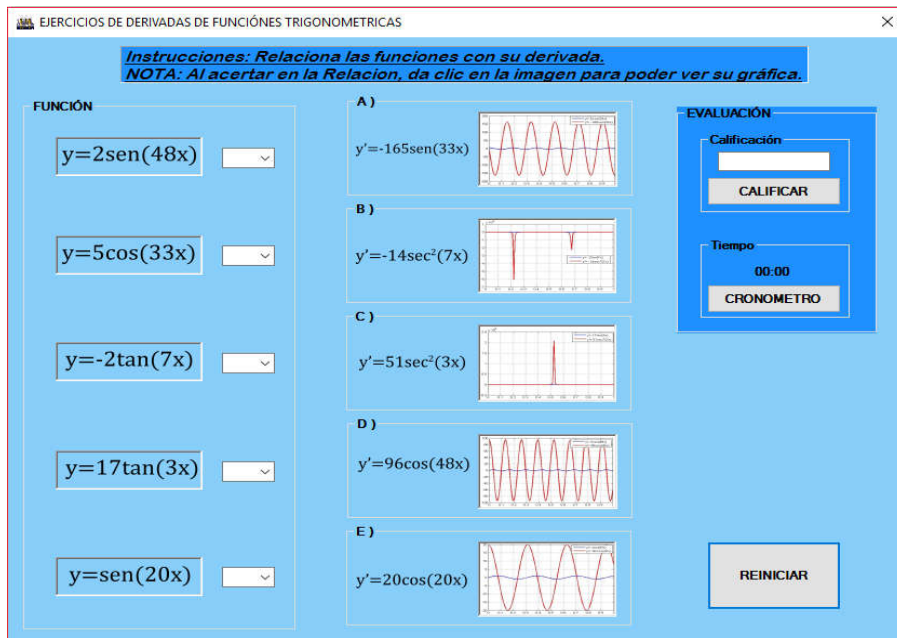


Fig.37.-Ventana de Ejercicios de funciones trigonométricas.



- Código de programación en botón :

CALIFICAR

```
PublicClassForm1
Dim seg, min AsInteger

PrivateSub CALIFICAR_Click(sender AsObject, e AsEventArgs) Handles CALIFICAR.Click
Dim SUMS AsInteger = 0

IfVal(ComboBox1.SelectedIndex) = 3 Then
    ComboBox1.BackColor = Color.Gold
    GroupBox5.BackColor = Color.Gold
    ComboBox1.ForeColor = Color.Black
    SUMS = SUMS + 2
    PictureBox12.Enabled = True
Else
    ComboBox1.BackColor = Color.Red
    GroupBox5.BackColor = Color.Red
    ComboBox1.ForeColor = Color.White
    PictureBox12.Enabled = False

EndIf

IfVal(ComboBox2.SelectedIndex) = 0 Then
    ComboBox2.BackColor = Color.Gold
    GroupBox2.BackColor = Color.Gold
    ComboBox2.ForeColor = Color.Black
    SUMS = SUMS + 2
    PictureBox7.Enabled = True
Else
    ComboBox2.BackColor = Color.Red
    GroupBox2.BackColor = Color.Red
    ComboBox2.ForeColor = Color.White
    PictureBox7.Enabled = False

EndIf

IfVal(ComboBox3.SelectedIndex) = 1 Then
    ComboBox3.BackColor = Color.Gold
    GroupBox3.BackColor = Color.Gold
    ComboBox3.ForeColor = Color.Black
    SUMS = SUMS + 2
    PictureBox8.Enabled = True
Else
    ComboBox3.BackColor = Color.Red
    GroupBox3.BackColor = Color.Red
    ComboBox3.ForeColor = Color.White
    PictureBox8.Enabled = False

EndIf

IfVal(ComboBox4.SelectedIndex) = 2 Then
    ComboBox4.BackColor = Color.Gold
    GroupBox4.BackColor = Color.Gold
    ComboBox4.ForeColor = Color.Black
    SUMS = SUMS + 2
    PictureBox10.Enabled = True
Else
    ComboBox4.BackColor = Color.Red
    GroupBox4.BackColor = Color.Red
    ComboBox4.ForeColor = Color.White
    PictureBox10.Enabled = False

EndIf

IfVal(ComboBox5.SelectedIndex) = 4 Then
```



```
ComboBox5.BackColor = Color.Gold
GroupBox6.BackColor = Color.Gold
ComboBox5.ForeColor = Color.Black
SUMS = SUMS + 2
PictureBox14.Enabled = True

Else
    ComboBox5.BackColor = Color.Red
    GroupBox6.BackColor = Color.Red
    ComboBox5.ForeColor = Color.White
    PictureBox14.Enabled = False

EndIf
    TextBox1.Text = SUMS
Timer1.Stop()
SelectCase SUMS
Case 0 To 5
    PictureBox17.Visible = True
    REPRO.Show()
Case 6 To 10
    PictureBox16.Visible = True
    GANASTE.Show()
EndSelect
EndSub
```

- Código de programación en botón :

CRONOMETRO

```
PrivateSub Button2_Click(sender AsObject, e AsEventArgs) Handles Button2.Click
    seg = 0
    min = 0
    Label7.Text = "00:00"
    Timer1.Start()
EndSub
```

- Código de programación en Timer :

 Timer1

```
PrivateSub Timer1_Tick(sender AsObject, e AsEventArgs) Handles Timer1.Tick
    seg += 1
    If seg = 60 Then
        min += 1
        seg = 0
    EndIf
    If min < 10 And seg < 10 Then
        Label7.Text = "0"& min & ":0"& seg
    ElseIf min < 10 And seg > 10 Then
        Label7.Text = "0"& min & ":"& seg
    ElseIf min > 10 And seg > 10 Then
        Label7.Text = "0"& min & ":"& seg
    ElseIf min > 10 And seg < 10 Then
        Label7.Text = min & ":"& seg
    EndIf
EndSub
```



- Código de programación en botón :

REINICIAR

```
PrivateSub Button1_Click(sender AsObject, e AsEventArgs) Handles Button1.Click
    ComboBox1.Text = Nothing : ComboBox2.Text = Nothing : ComboBox3.Text = Nothing :
    ComboBox4.Text = Nothing : ComboBox5.Text = Nothing
    ComboBox1.Focus()
    ComboBox1.BackColor = Color.White
    ComboBox1.ForeColor = Color.Black
    ComboBox2.BackColor = Color.White
    ComboBox2.ForeColor = Color.Black
    ComboBox3.BackColor = Color.White
    ComboBox3.ForeColor = Color.Black
    ComboBox4.BackColor = Color.White
    ComboBox4.ForeColor = Color.Black
    ComboBox5.BackColor = Color.White
    ComboBox5.ForeColor = Color.Black
    ComboBox5.BackColor = Color.White
    GroupBox2.BackColor = Color.LightSkyBlue
    GroupBox3.BackColor = Color.LightSkyBlue
    GroupBox4.BackColor = Color.LightSkyBlue
    GroupBox5.BackColor = Color.LightSkyBlue
    GroupBox6.BackColor = Color.LightSkyBlue
    TextBox1.Text = Nothing
    Timer1.Stop()
    Label7.Text = "00:00"
    PictureBox17.Visible = False
    PictureBox16.Visible = False
EndSub
```

- ▢ Código de programación para las ventanas emergentes donde aparecen los mensajes de aprobación en todos ejercicios :

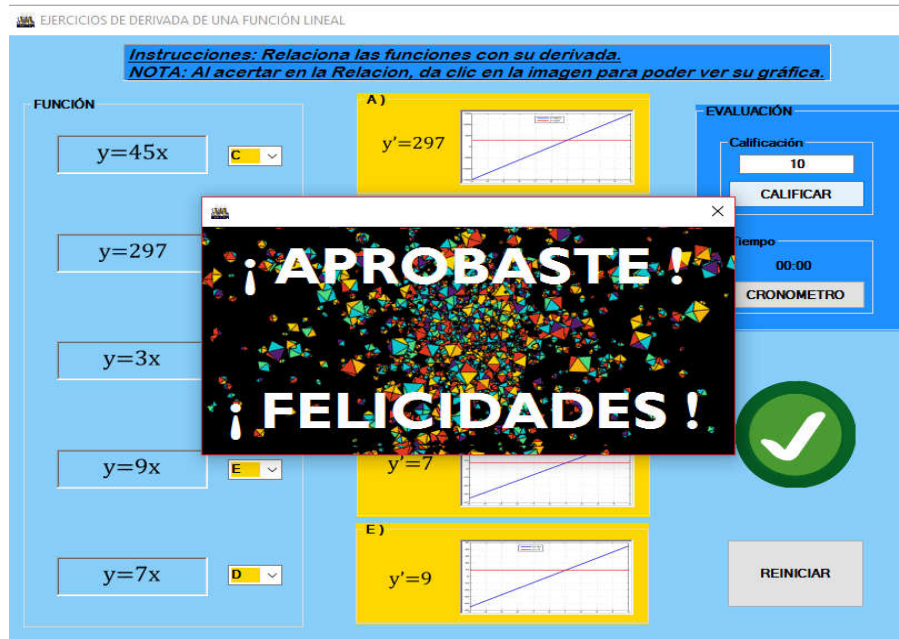


Fig.38.-Ventana de mensaje emergente de aprobación.



○ Código de programación en Timer :



```
PublicClassGANASTE
```

```
PrivateSub Timer1_Tick(sender AsObject, e AsEventArgs) Handles Timer1.Tick
```

```
Static C AsInteger
```

```
    C = C + 1 'Inicializando
```

```
If C = 1 Then
```

```
    Label2.ForeColor = Color.Green
```

```
ElseIf C = 2 Then
```

```
    Label2.ForeColor = Color.Yellow
```

```
ElseIf C = 3 Then
```

```
    Label2.ForeColor = Color.Aqua
```

```
ElseIf C = 4 Then
```

```
    Label2.ForeColor = Color.Red
```

```
ElseIf C = 5 Then
```

```
    Label2.ForeColor = Color.White
```

```
Else : C = 6
```

```
    C = 0
```

```
EndIf
```

```
EndSub
```

```
EndClass
```

- ▢ Código de programación para las ventanas emergentes donde aparecen los mensajes de no aprobación en todos ejercicios:

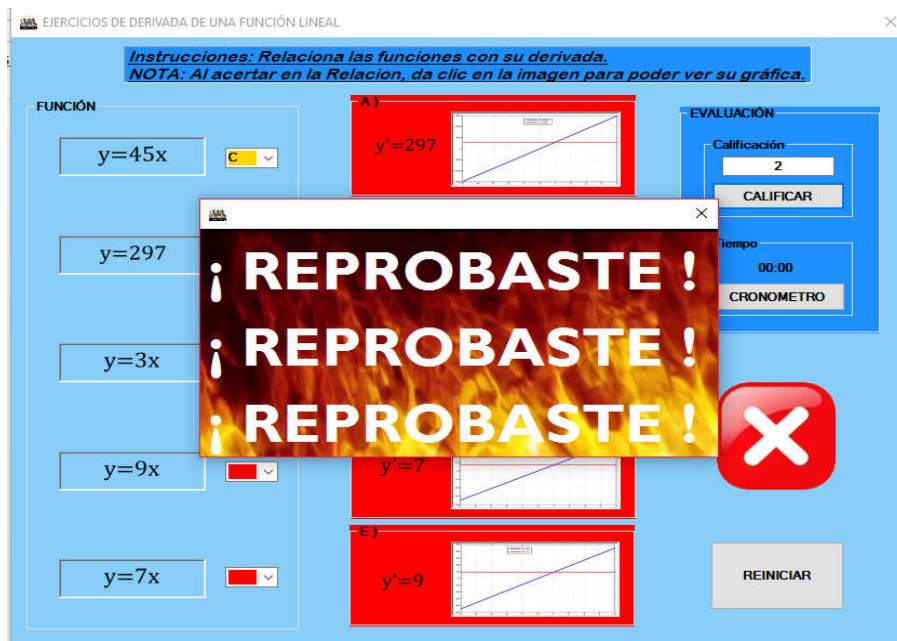
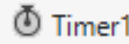


Fig.39.-Ventana de mensaje emergente de reprobación.



○ Código de programación en Timer :



```
PublicClassREPRO
PrivateSub Timer1_Tick(sender AsObject, e AsEventArgs) Handles Timer1.Tick
Static C AsInteger
    C = C + 1 'Inicializando
If C = 1 Then
    Label2.ForeColor = Color.Blue
ElseIf C = 2 Then
    Label2.ForeColor = Color.DeepPink
ElseIf C = 3 Then
    Label2.ForeColor = Color.Aqua
ElseIf C = 4 Then
    Label2.ForeColor = Color.Red
ElseIf C = 5 Then
    Label2.ForeColor = Color.White
Else : C = 6
    C = 0
EndIf
EndSub
EndClass
```

3.3.3 Códigos de programación para las ventanas de conceptos de Learn Math.

Las ventanas para conceptos propuestos en Learn Math, fueron realizados en Visual Studio 2012 ya es un programa bastante amigable y más estético para colocar texto que es parte primordial de las ventanas de conceptos.

📄 Códigos de programación para ventanas de conceptos de funciones constantes y su derivada en Learn Math.

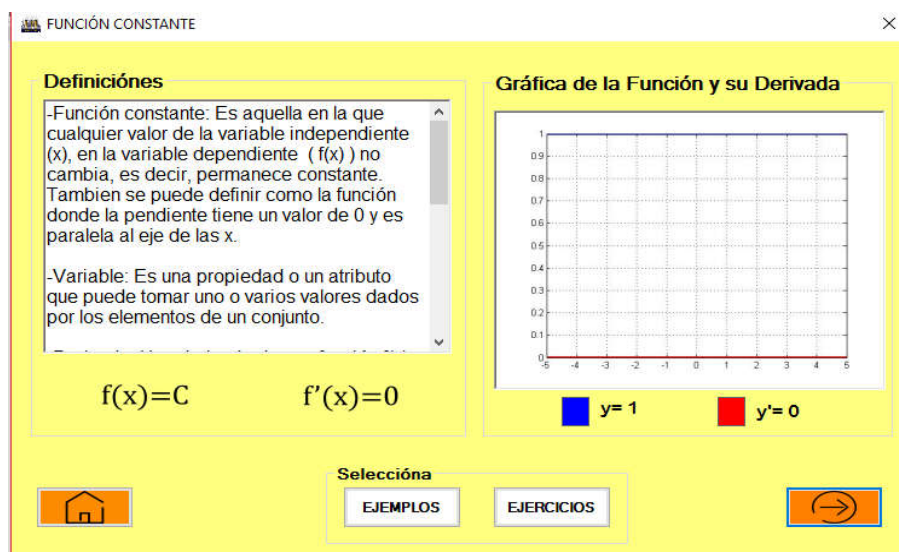


Fig.40.-Ventana conceptos de función constante.



- **Código de programación en botón :** 

```
PrivateSub Button4_Click(sender AsObject, e AsEventArgs) Handles Button4.Click  
Me.Hide()  
Form1.Show()  
EndSub
```

- **Código de programación en botón :** 

```
PrivateSub Button3_Click(sender AsObject, e AsEventArgs) Handles Button3.Click  
FUNCION_LINEAL.Show()  
Me.Hide()  
EndSub  
EndClass
```

- **Código de programación en botón :** 

```
PrivateSub Button1_Click(sender AsObject, e AsEventArgs) Handles Button1.Click  
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\MLAB\DC.exe")  
EndSub
```

- **Código de programación en botón :** 

```
PrivateSub Button2_Click(sender AsObject, e AsEventArgs) Handles Button2.Click  
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\VB\DC.exe") EndSub
```



- Códigos de programación para ventanas de conceptos de funciones lineales y su derivada en Learn Math.

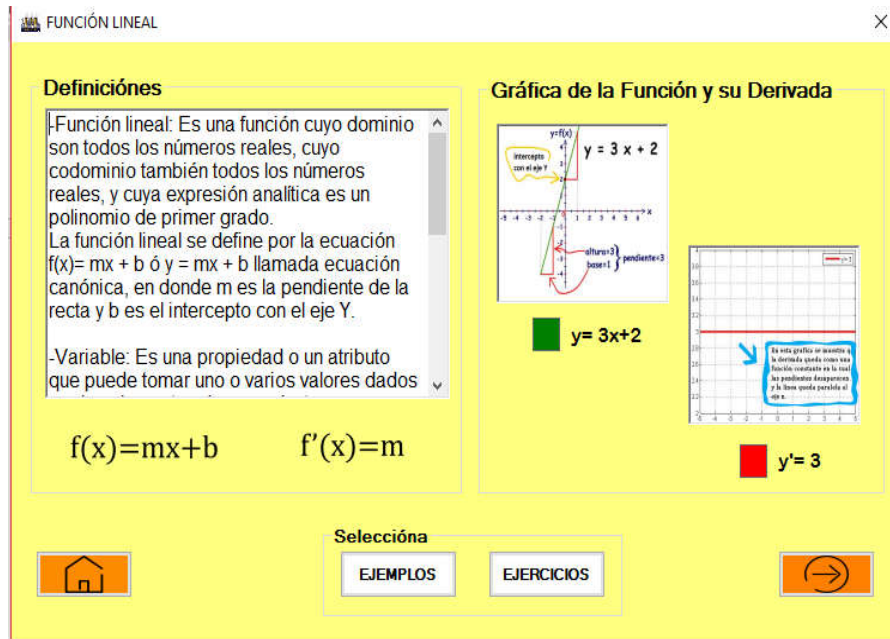


Fig.41.-Ventana conceptos de función lineal.

- Código de programación en botón : 

```
PrivateSub Button4_Click(sender AsObject, e AsEventArgs) Handles Button4.Click  
Me.Hide()  
Form1.Show()  
EndSub
```

- Código de programación en botón : 

```
PrivateSub Button3_Click(sender AsObject, e AsEventArgs) Handles Button3.Click  
FUNCION_LINEAL.Show()  
Me.Hide()  
EndSub  
EndClass
```



- Código de programación en botón : **EJEMPLOS**

```
PrivateSub Button1_Click(sender AsObject, e AsEventArgs) Handles Button1.Click  
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\MLAB\DLIN.exe")  
EndSub
```

- Código de programación en botón : **EJERCICIOS**

```
PrivateSub Button2_Click(sender AsObject, e AsEventArgs) Handles Button2.Click  
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\VB\DLIN.exe")  
EndSub  
EndClass
```

- 📄 Códigos de programación para ventanas de conceptos de funciones polinomiales y su derivada en Learn Math.

Definiciones

-Función polinomial: es una función en que $f(x)$ es un polinomio en x . Una función polinomial de grado n es escrita como . Las funciones polinomiales están definidas y son continuas en todos los números reales.

-Variable: Es una propiedad o un atributo que puede tomar uno o varios valores dados por los elementos de un conjunto.

-Derivada: Una derivada de una función $f(x)$ en un punto x_0 se obtiene comparando el

$f(x)=ax^n+bx^{n-1}...$ $f'(x)=nax^{n-1}+(n-1)bx^{(n-1)-1}..$

Gráfica de la Función y su Derivada

$y=x^3$

$y'=3x^2$

Selección

EJEMPLOS EJERCICIOS

Fig.42.-Ventana conceptos de función polinomial.



- **Código de programación en botón :** 

```
PrivateSub Button4_Click(sender AsObject, e AsEventArgs) Handles Button4.Click  
Me.Hide()  
Form1.Show()  
EndSub
```

- **Código de programación en botón :** 

```
PrivateSub Button3_Click(sender AsObject, e AsEventArgs) Handles Button3.Click  
FUNCION_LINEAL.Show()  
Me.Hide()  
EndSub  
EndClass
```

- **Código de programación en botón :** 

```
PrivateSub Button1_Click(sender AsObject, e AsEventArgs) Handles Button1.Click  
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\MLAB\DPOL.exe")  
EndSub
```

- **Código de programación en botón :** 

```
PrivateSub Button2_Click(sender AsObject, e AsEventArgs) Handles Button2.Click  
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\VB\DPOL.exe")EndSub
```



- Códigos de programación para ventanas de conceptos de funciones logarítmicas y su derivada en Learn Math.

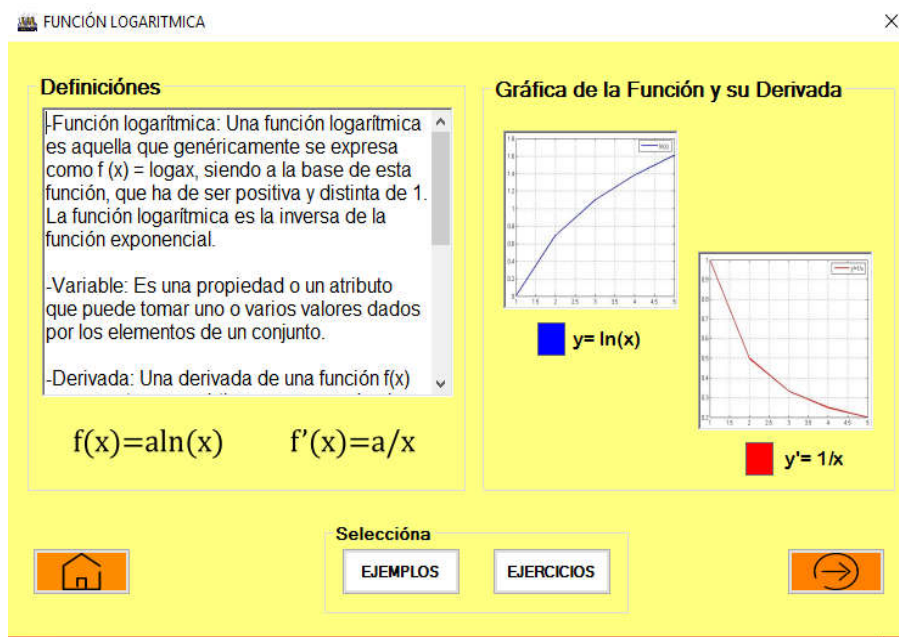


Fig.43.-Ventana conceptos de función logarítmica.

- Código de programación en botón : 

```
PrivateSub Button4_Click(sender AsObject, e AsEventArgs) Handles Button4.Click  
Me.Hide()  
Form1.Show()  
EndSub
```

- Código de programación en botón : 

```
PrivateSub Button3_Click(sender AsObject, e AsEventArgs) Handles Button3.Click  
FUNCION_LINEAL.Show()  
Me.Hide()  
EndSub  
EndClass
```



- Código de programación en botón : EJEMPLOS

```
PrivateSub Button1_Click(sender AsObject, e AsEventArgs) Handles Button1.Click
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\MLAB\DLOG.exe")
EndSub
```

- Código de programación en botón : EJERCICIOS

```
PrivateSub Button2_Click(sender AsObject, e AsEventArgs) Handles Button2.Click
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\VB\DLOG.exe")
EndSub
```

- 📄 Códigos de programación para ventanas de conceptos de funciones trigonométricas y su derivada en Learn Math.

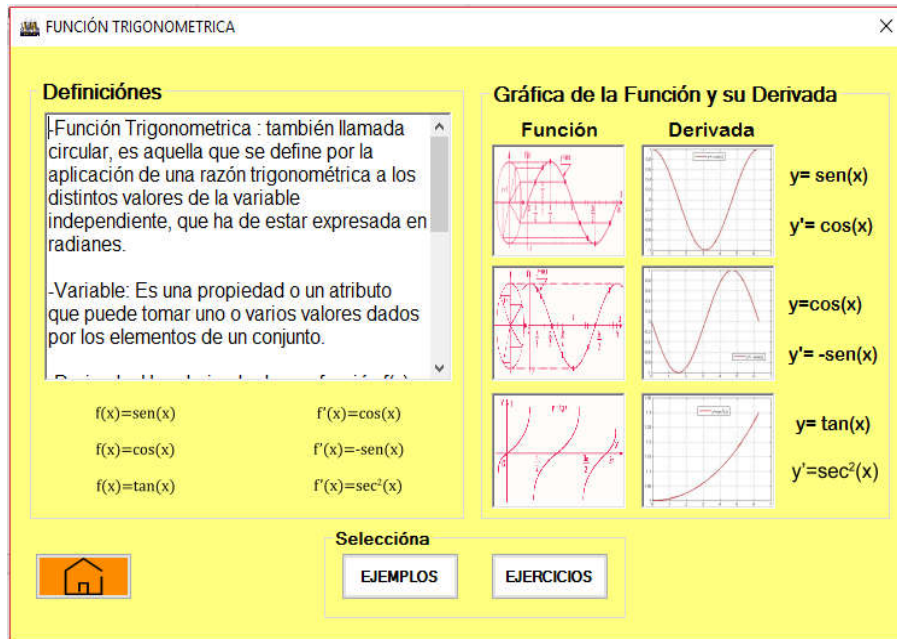


Fig.44.-Ventana conceptos de función trigonométrica.

- Código de programación en botón :

```
PrivateSub Button4_Click(sender AsObject, e AsEventArgs) Handles Button4.Click
Me.Hide()
```




Form1.Show()
EndSub

- Código de programación en botón : **EJEMPLOS**

```
PrivateSub Button1_Click(sender AsObject, e AsEventArgs) Handles Button1.Click
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\MLAB\DTRI.exe")
EndSub
```

- Código de programación en botón : **EJERCICIOS**

```
PrivateSub Button2_Click(sender AsObject, e AsEventArgs) Handles Button2.Click
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\VB\DTRI.exe")
EndSub
```

3.3.4 Códigos de programación para las ventanas de inicio para Learn Math.

Las ventanas para conceptos propuestos en Learn Math, fueron realizadas en Visual Studio 2012 ya que es un programa bastante amigable y más estético con base en la apariencia en las ventanas, así también para colocar texto y hacer presentaciones de inicio de programas.

Se crea en Visual Studio un SplashScreen para dar un el efecto de entrada y la transparencia.

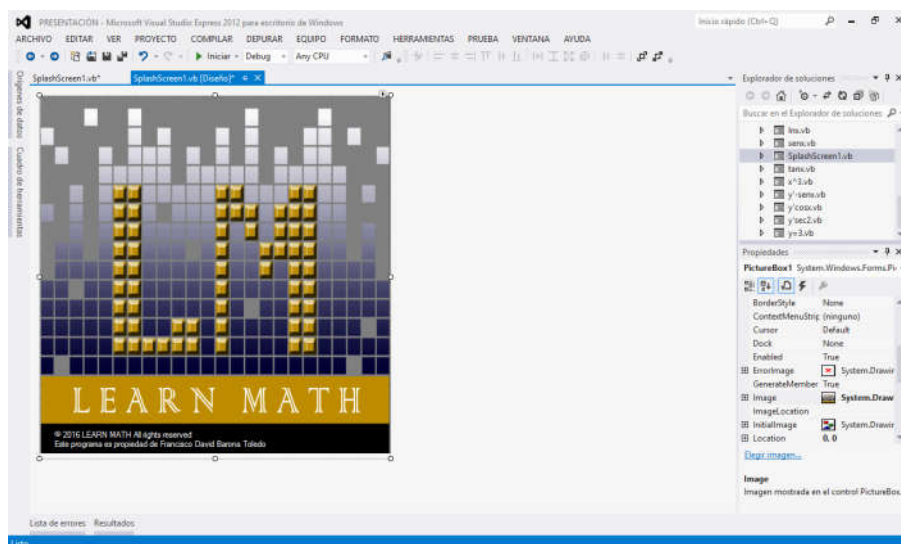


Fig.45.-Creacion de SplashScreen en Visual Studio 2012.



Como siguiente paso, se creará una ventana con un menú de inicio, este menú de inicio contendrá botones para llamar las demás aplicaciones.

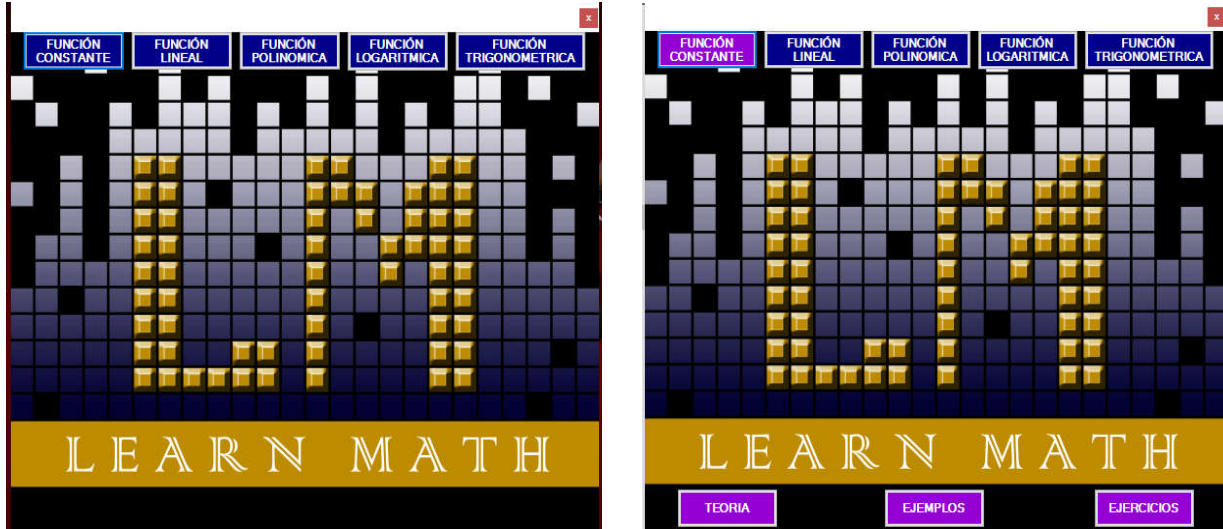


Fig.46.-Creacion de menú de inicio para Learn Math.

- Código de programación en botón :



```
PrivateSub Button1_Click(sender AsObject, e AsEventArgs) Handles Button1.Click
    Button1.BackColor = Color.DarkViolet
    Button2.BackColor = Color.DarkBlue
    Button3.BackColor = Color.DarkBlue
    Button4.BackColor = Color.DarkBlue
    Button5.BackColor = Color.DarkBlue
    Button6.Visible = True
    Button6.BackColor = Color.DarkViolet
    Button7.Visible = True
    Button7.BackColor = Color.DarkViolet
    Button8.Visible = True
    Button8.BackColor = Color.DarkViolet
    Button9.Visible = False
    Button10.Visible = False
    Button11.Visible = False
    Button12.Visible = False
    Button13.Visible = False
    Button14.Visible = False
    Button15.Visible = False
    Button16.Visible = False
    Button17.Visible = False
    Button18.Visible = False
    Button19.Visible = False
    Button20.Visible = False
EndSub
```



○ **Código de programación en botón :**

TEORIA

```
PrivateSub Button20_Click(sender AsObject, e AsEventArgs) Handles Button20.Click
FUNCIÓN_Constante.Show()
Me.Hide()
EndSub
```

○ **Código de programación en botón :**

EJEMPLOS

```
PrivateSub Button7_Click(sender AsObject, e AsEventArgs) Handles Button7.Click
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\MLAB\DC.exe")
EndSub
```

○ **Código de programación en botón :**

EJERCICIOS

```
PrivateSub Button6_Click(sender AsObject, e AsEventArgs) Handles Button6.Click
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\VB\DC.exe") EndSub
```

○ **Código de programación en botón :**

FUNCIÓN
LINEAL

```
PrivateSub Button2_Click(sender AsObject, e AsEventArgs) Handles Button2.Click
Button1.BackColor = Color.DarkBlue
Button2.BackColor = Color.DarkGreen
Button3.BackColor = Color.DarkBlue
Button4.BackColor = Color.DarkBlue
Button5.BackColor = Color.DarkBlue
Button6.Visible = False
Button7.Visible = False
Button8.Visible = False
Button9.Visible = True
Button9.BackColor = Color.DarkGreen
Button10.Visible = True
Button10.BackColor = Color.DarkGreen
Button11.Visible = True
Button11.BackColor = Color.DarkGreen
Button12.Visible = False
Button13.Visible = False
```



```
Button14.Visible = False  
Button15.Visible = False  
Button16.Visible = False  
Button17.Visible = False  
Button18.Visible = False  
Button19.Visible = False  
Button20.Visible = False
```

EndSub

- Código de programación en botón :

TEORIA

```
PrivateSub Button9_Click(sender AsObject, e AsEventArgs) Handles Button9.Click  
FUNCION_LINEAL.Show()  
Me.Hide()  
EndSub
```

- Código de programación en botón :

EJEMPLOS

```
PrivateSub Button10_Click(sender AsObject, e AsEventArgs) Handles Button10.Click  
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\MLAB\DLIN.exe")EndSub
```

- Código de programación en botón :

EJERCICIOS

```
PrivateSub Button11_Click(sender AsObject, e AsEventArgs) Handles Button11.Click  
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\VB\DLIN.exe")  
EndSub
```

- Código de programación en botón :

FUNCIÓN
POLINOMICA

```
PrivateSub Button3_Click(sender AsObject, e AsEventArgs) Handles Button3.Click  
  
Button3.BackColor = Color.DarkOrange  
Button1.BackColor = Color.DarkBlue  
Button2.BackColor = Color.DarkBlue  
Button4.BackColor = Color.DarkBlue  
Button5.BackColor = Color.DarkBlue
```



```
Button6.Visible = False
Button7.Visible = False
Button8.Visible = False
Button9.Visible = False
Button10.Visible = False
Button11.Visible = False
Button12.Visible = True
Button12.BackColor = Color.DarkOrange
Button13.Visible = True
Button13.BackColor = Color.DarkOrange
Button14.Visible = True
Button14.BackColor = Color.DarkOrange
Button15.Visible = False
Button16.Visible = False
Button17.Visible = False
Button18.Visible = False
Button19.Visible = False
Button20.Visible = False
```

EndSub

- Código de programación en botón :

TEORIA

```
PrivateSub Button12_Click(sender AsObject, e AsEventArgs) Handles Button12.Click
FUNCION_POLINOMIAL.Show()
Me.Hide()
EndSub
```

- Código de programación en botón :

EJEMPLOS

```
PrivateSub Button13_Click(sender AsObject, e AsEventArgs) Handles Button13.Click
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\MLAB\DPOL.exe")
EndSub
```

- Código de programación en botón :

EJERCICIOS

```
PrivateSub Button14_Click(sender AsObject, e AsEventArgs) Handles Button14.Click
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\VB\DPOL.exe")
EndSub
```

- Código de programación en botón :

FUNCIÓN
LOGARITMICA

```
PrivateSub Button4_Click(sender AsObject, e AsEventArgs) Handles Button4.Click
Button1.BackColor = Color.DarkBlue
Button2.BackColor = Color.DarkBlue
```



```
Button3.BackColor = Color.DarkBlue
Button4.BackColor = Color.DarkRed
Button6.Visible = False
Button7.Visible = False
Button8.Visible = False
Button9.Visible = False
Button10.Visible = False
Button11.Visible = False
Button12.Visible = False
Button13.Visible = False
Button14.Visible = False
Button15.Visible = True
Button15.BackColor = Color.DarkRed
Button16.Visible = True
Button16.BackColor = Color.DarkRed
Button17.Visible = True
Button17.BackColor = Color.DarkRed
Button18.Visible = False
Button19.Visible = False
Button20.Visible = False
```

EndSub

○ Código de programación en botón : **TEORIA**

```
PrivateSub Button17_Click(sender AsObject, e AsEventArgs) Handles Button17.Click
FUNCIÓN_LOGARITMICA.Show()
Me.Hide()
EndSub
```

○ Código de programación en botón : **EJEMPLOS**

```
PrivateSub Button16_Click(sender AsObject, e AsEventArgs) Handles Button16.Click
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\MLAB\DLOG.exe")
EndSub
```

○ Código de programación en botón : **EJERCICIOS**

```
PrivateSub Button15_Click(sender AsObject, e AsEventArgs) Handles Button15.Click
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\VB\DLOG.exe")
EndSub
```

○ Código de programación en botón : **FUNCIÓN TRIGONOMETRICA**

```
PrivateSub Button5_Click(sender AsObject, e AsEventArgs) Handles Button5.Click
Button1.BackColor = Color.DarkBlue
Button2.BackColor = Color.DarkBlue
```



```
Button3.BackColor = Color.DarkBlue
Button4.BackColor = Color.DarkBlue
Button5.BackColor = Color.DarkTurquoise
Button6.Visible = False
Button7.Visible = False
Button8.Visible = False
Button9.Visible = False
Button10.Visible = False
Button11.Visible = False
Button12.Visible = False
Button13.Visible = False
Button14.Visible = False
Button15.Visible = False
Button16.Visible = False
Button17.Visible = False
Button18.Visible = True
Button18.BackColor = Color.DarkTurquoise
Button19.Visible = True
Button19.BackColor = Color.DarkTurquoise
Button20.Visible = True
Button20.BackColor = Color.DarkTurquoise
```

EndSub

- Código de programación en botón :

TEORIA

```
PrivateSub Button20_Click(sender AsObject, e AsEventArgs) Handles Button20.Click
FUNCIÓN_TRIGONOMETRICA.Show()
Me.Hide()
EndSub
```

- Código de programación en botón :

EJEMPLOS

```
PrivateSub Button19_Click(sender AsObject, e AsEventArgs) Handles Button19.Click
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\MLAB\DTRI.exe")
EndSub
```

- Código de programación en botón :

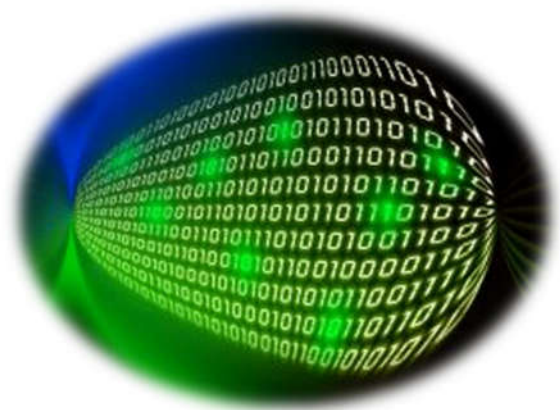
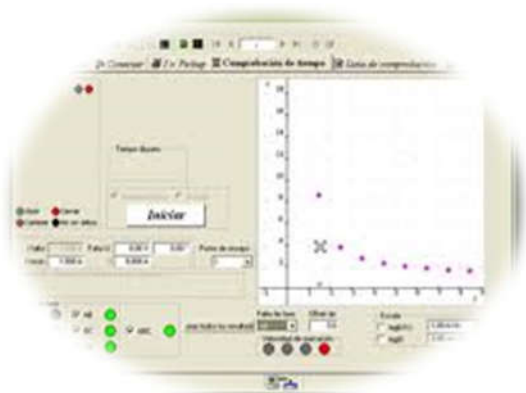
EJERCICIOS

```
PrivateSub Button18_Click(sender AsObject, e AsEventArgs) Handles Button18.Click
System.Diagnostics.Process.Start("C:\Program Files (x86)\Learn Math\VB\DTRI.exe")
EndSub
```



CAPÍTULO 4

PRUEBAS DEL MATERIAL DIDÁCTICO.





4.1 Pruebas de ejecución del programa.

Durante el proceso de elaboración del programa Learn Math se fueron haciendo pruebas de ejecución del programa, esto para verificar que no hubiera errores al momento de ejecutarlo. En este proceso si, hubo problemas en el código de programación, principalmente en el código de Matlab R2013a ya que es más complejo en cuanto a la programación y a la creación de programas ejecutables y estos errores no permitían que el programa se ejecutara de manera correcta, sin embargo este tipo de problemas fueron solucionados por lo cual el programa ahora puede ejecutarse de una manera segura y sin errores.

4.2 Pruebas del programa en distintas versiones de Windows ejecución del programa.

Ya que se tenía el programa completo, se realizaron pruebas para verificar que el programa se ejecutara en diferentes versiones de Windows, esto para asegurarse de que se podía utilizar en cualquier versión de Windows sin embargo, lamentablemente el programa sólo se ejecuta de manera correcta en versiones de Windows 8, y Windows 10 de 64 bits ya que en las otras versiones el programa se ejecuta pero al momento de manipular las aplicaciones manda un mensaje de error por lo cual las únicas versiones seleccionadas fueron las ya antes mencionadas, además de que en el programa, sus gráficos están configurados con el ambiente de Windows 10, por este motivo en otras versiones las imágenes pueden llegar a verse mal o los colores pueden distorsionarse debido a que los gráficos de estos programas son más avanzados por lo cual necesitan una tarjeta de gráficos más actual, cosa que equipos de cómputo ya pasados no soportan en la actualidad.



ANÁLISIS DE RESULTADOS

Se obtuvo un programa que cumple con el contenido de un material didáctico cuyas características son:

- ✓ Facilidad de uso.
- ✓ Uso individual o colectivo.
- ✓ Versatilidad.
- ✓ Abiertos.
- ✓ Proporcionar información.
- ✓ Capacidad de motivación.
- ✓ Adecuación al ritmo de trabajo de los alumnos.
- ✓ Estimular
- ✓ Esfuerzo cognitivo.
- ✓ Disponibilidad.

No obstante, al material aún se le pueden hacer mejoras ya que esta es la versión inicial del material didáctico.

Se presentó a varias personas entre ellas estudiantes de la carrera de Ingeniería Química y profesores de la misma carrera junto con el pizarrón interactivo y les pareció una idea innovadora y bastante atractiva.

El programa propuesto tuvo limitantes en su programación ya que no se pueden declarar variables simbólicas en Matlab R2013a y crear un ejecutable esto con el fin de programar de una mejor manera por lo que se le tuvieron que hacer arreglos numéricos para que el programa pudiera ejecutarse de la manera correcta y sin errores.



CONCLUSIONES

En conclusión, esta propuesta puede ser una herramienta útil para el profesor y para el estudiante. El primero reduciría el tiempo de escritura y dibujo en su exposición y el segundo tendría la explicación en un archivo portátil. Con ello, se pretende confirmar que las personas aprenden más de una manera mas grafica y visual. Sin embargo, no se busca reemplazar a un profesor o a un libro, lo que se busca es ajustarse a nuevos métodos de enseñanza conforme a la época de los nuevos estudiantes y ocupar la tecnología en la enseñanza, algo que pueda hacer que el estudiante crezca de manera profesional utilizando el uso de las tecnologías a su favor. En otros países están invirtiendo mucho en tecnología para hacer más competitivos a sus estudiantes. En la vida laboral es un requisito tener conocimientos tecnológicos y las empresas atesoran a los empleados productivos. Al comenzar con un tema del plan de estudios el empleo de los recursos digitales en el aula se espera que los resultados sean benéficos para la carrera.



REFERENCIAS BIBLIOGRÁFICAS

1. Purcell, E.J., Varberg, D. y Rigdon, S.E. (2007). Cálculo diferencial e integral, 9° ed. México: Prentice Hall.
2. Stewart, J. (2007). Precalculo. 5° ed. Thomson.
3. Stewart, J. (2009). Calculus: concepts and contexts. 4° ed. Cengage Learning.
4. Stewart, J. (2008). Cálculo trascendente temprano, 6° ed. Thomson.
5. Zill, D.G. (1992). Calculus. 3° ed. Editorial I.T.P
6. <http://www.studygs.net/espanol/visual.htm>. [Revisado el 10 Febrero 2016]
7. <http://definicion.de/matematicas/#ixzz40w0N8Nqk>[Revisado el 10 Febrero 2016].
8. <http://oscar-calculodiferencial.blogspot.mx/2011/09/definicion-de-calculo-diferencial.html> [Revisado el 15 Febrero 2016].
9. <http://www.numerosreales.com/>[Revisado el 15 Febrero 2016].
10. <http://deconceptos.com/matematica/variable>[Revisado el 20 Febrero 2016].
11. <http://deconceptos.com/general/funciones#ixzz40ewalP1m>[Revisado el 20 Febrero 2016].
12. http://www.eis.uva.es/reic/jc/IQweb/calculo_diferencial.htm[Revisado el 20 Febrero 2016].
13. http://www.abc.es/sociedad/abci-10-tendencias-revolucionar-sistema-educativo-201602110319_noticia.html [Revisado el 20 Febrero 2016].
14. <http://iesainformatica66.blogspot.mx/2011/04/tecnicas-visuales-de-aprendizaje.html> [Revisado el 20 Febrero 2016].
15. Rubén Flores E., Marco Antonio Valencia A., Guillermo Dávila R., Martín G. García A. Fundamentos del cálculo. Proyecto FOMIX. Editorial GARABATOS.
16. http://www.iboard.com.mx/catalogo_foto_descripcion_opciones.php?categoria=1&subcategoria=1[Revisado el 23 Febrero 2016].
17. <http://www.colombiaaprende.edu.co/html/directivos/1598/article-172365.html>[Revisado el 23 Febrero 2016]
18. Definición de material didáctico - Qué es, Significado y Concepto <http://definicion.de/material-didactico/#ixzz4GmFKe4uo> [Revisado el 08 de Agosto del 2016]